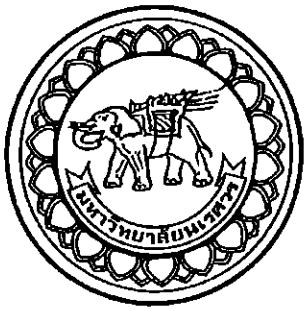


การหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก

A METAHEURISTIC FOR CAPACITATED VEHICLE ROUTING PROBLEM

นางสาวปิ่นนภา เกตุศรี รหัส 52360386

นางสาวสุวรรณา บุญชุ่ม รหัส 52360720

ห้องสมุด คณะวิศวกรรมศาสตร์
วันที่รับ 24 ก.ค. 2556
เลขทะเบียน 16317065
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร ป๒๑๗ ๒๕๕๖

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน การหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก

ผู้ดำเนินโครงงาน	นางสาวปิ่นนภา เกตุศรี	รหัส 52360386
	นางสาวสุวรรณา บุญชุ่ม	รหัส 52360720
ที่ปรึกษาโครงงาน	ดร.ขวัญนิธิ คำเมือง	
สาขาวิชา	วิศวกรรมอุตสาหการ	
ภาควิชา	วิศวกรรมอุตสาหการ	
ปีการศึกษา	2555	

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ

.....ที่ปรึกษาโครงงาน
(ดร. ขวัญนิธิ คำเมือง)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. ภูพงษ์ พงษ์เจริญ)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ศรีสัจจา วิทยศักดิ์)

ชื่อหัวข้อโครงงาน	การหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก
ผู้ดำเนินโครงงาน	นางสาวปิ่นนภา เกตุศรี รหัส 52360386
	นางสาวสุวรรณา บุญชุ่ม รหัส 52360720
ที่ปรึกษาโครงงาน	ดร.ขวัญนิธิ คำเมือง
สาขาวิชา	วิศวกรรมอุตสาหการ
ภาควิชา	วิศวกรรมอุตสาหการ
ปีการศึกษา	2555

.....

บทคัดย่อ

โครงงานวิจัยนี้จะเป็นจะเป็นการสร้างวิธีการในการจัดเรียงยานพาหนะในการขนส่งสินค้าให้กับลูกค้า โดยวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ มาช่วยในการจัดเรียงยานพาหนะ รวมทั้งได้มีการพัฒนาและสร้างโปรแกรมเพื่อใช้ในการแก้ไขปัญหา ซึ่งจะใช้การเขียนโปรแกรมโดยใช้ภาษา Visual Basic for Application บน Microsoft Excal

การดำเนินการออกแบบจำลองนี้ จะมีกระบวนการหลักอยู่ 4 กระบวนการ คือการหาคำตอบแรกที่ได้จากการสุ่ม การหาคำตอบใหม่ในย่านคำตอบเดิม กระบวนการจัดเรียงรอบรถ และออกแบบเมตาฮิวริสติก ซึ่งในการทดสอบประสิทธิภาพของการจัดเรียงยานพาหนะโดยวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ ได้ทำการทดสอบกับปัญหาขนาดต่างๆ และมีการเปรียบเทียบผลลัพธ์ที่ได้กับงานวิจัยของผู้ทดลองเอง จากการศึกษาสามารถสรุปได้ว่า หลักการแบบสลับ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 เป็นหลักการที่ควรนำมาใช้ในการหาคำตอบมากกว่าหลักการแบบอื่น เนื่องจากเป็นหลักการที่ให้ค่าคำตอบและเวลาที่ใช้ในการประมวลผลที่ดีกว่าแบบอื่น

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยความอนุเคราะห์ของ บุคคลหลายท่าน ซึ่งผู้มีพระคุณท่านแรกที่ผู้ศึกษาใคร่ขอกราบพระคุณ คือ ดร.ขวัญนิธิ คำเมือง อาจารย์ผู้สอนที่ได้ให้ความรู้ คำแนะนำ ตรวจทาน และแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่ทุกขั้นตอน เพื่อให้การเขียนรายงานค้นคว้าฉบับนี้สมบูรณ์ที่สุด ท่านที่สอง คือ อาจารย์กานต์ ลี่วัฒนายิ่งยง ที่ให้คำแนะนำตรวจทาน และแก้ไขข้อบกพร่องต่างๆ นอกจากนี้ ผู้ศึกษาใคร่ขอขอบพระคุณ คณะกรรมการทั้งสามท่าน อันประกอบไปด้วย ผศ.ดร. ภูพงษ์ พงษ์เจริญ อาจารย์ศรีสัจจา วิทยศักดิ์ และ ดร.สุธนิตย์ พุทธพนม ที่กรุณาให้ความช่วยเหลือในการแก้ไขและให้คำแนะนำที่มีประโยชน์ ที่มีส่วนทำให้งานวิจัยครั้งนี้มีคุณค่ามากยิ่งขึ้น จนสามารถออกมาเป็นปริญญานิพนธ์ฉบับนี้

อีกสิ่งหนึ่งที่จะลืมไม่ได้ก็ คือ ความกรุณาจาก บิดา มารดา พี่ และน้อง ที่อยู่เบื้องหลังในความสำเร็จนี้ ที่ได้สนับสนุนในเรื่องทุนการศึกษา และให้กำลังใจในทุกๆเรื่อง ตลอดมา ทางผู้ดำเนินโครงงานใคร่ขอกราบขอบพระคุณเป็นอย่างสูง ไว้ ณ ที่นี้ด้วย

คณะผู้ดำเนินโครงงานวิศวกรรม
นางสาวปิ่นนภา เกตุศรี
นางสาวสุวรรณา บุญซุ่ม

มกราคม 2556

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์.....ก
บทคัดย่อ.....ข
กิตติกรรมประกาศ.....ค
สารบัญ.....ง
สารบัญตาราง.....ช
สารบัญรูป.....ฌ

บทที่ 1 บทนำ.....1
1.1 ความเป็นมาและความสำคัญของโครงงาน.....1
1.2 วัตถุประสงค์ของโครงงาน.....1
1.3 เกณฑ์ชี้วัดผลงาน.....2
1.4 เกณฑ์ชี้วัดผลสำเร็จ.....2
1.5 ขอบเขตการดำเนินโครงงาน.....2
1.6 สถานที่ในการดำเนินโครงงาน.....2
1.7 ระยะเวลาในการดำเนินโครงงาน.....3
1.8 ขั้นตอนและแผนการดำเนินโครงงาน.....3

บทที่ 2 หลักการและทฤษฎี.....4
2.1 ปัญหาการจัดเส้นทางสำหรับยานพาหนะ (Vehicle Routing Problem : VRP).....4
2.2 ปัญหาการจัดเส้นทางยานพาหนะ.....4
2.2.1 กลุ่มลูกค้า (Set of Customer).....4
2.2.2 ยานพาหนะ (Vehicle).....4
2.2.3 คลังสินค้า (Depot).....5
2.2.4 เที่ยวรถ (Route).....5
2.3 รูปแบบของปัญหาการจัดเส้นทางสำหรับยานพาหนะ.....5
2.3.1 ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ (Capacitated Vehicle Routing Problem : CVRP).....5
2.3.2 ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบหลายคลังสินค้า (Multiple-Depots Vehicle Routing Problem : MDVRP).....6
2.3.3 ปัญหาการจัดเส้นทางสำหรับพาหนะแบบมีข้อจำกัดเนื่องจากเวลา (Vehicle Routing Problem with Time Windows : VRPTW).....6

สารบัญ (ต่อ)

หน้า

2.3.4 ปัญหาการจัดเส้นทางสำหรับพาหนะแบบมีข้อจำกัดเนื่องจากความไม่แน่นอน (Stochastic Vehicle Routing Problem : SVRP).....7
2.3.5 ปัญหาการจัดเส้นทางสำหรับพาหนะแบบมีการรับและส่งสินค้า (Vehicle Routing Problem with Pick – Up and Delivery : VRPPD).....7
2.4 ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ).....7
2.5 วิธีแก้ปัญหาในการหาค่าคำตอบที่ดีที่สุดของปัญหาการจัดเส้นทางสำหรับยานพาหนะ.....10
2.5.1 วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์.....10
2.5.2 วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณค่า.....11
2.6 วิธีฮิวริสติก (Heuristic Method).....11
2.6.1 การจัดเส้นทางสำหรับยานพาหนะแบบมีเงื่อนไขเนื่องจากความจุ ด้วยวิธีการเปรียบเทียบการประหยัด (Saving Algorithm).....11
2.6.2 ตัวอย่างการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีเงื่อนไขเนื่องจากความจุ ด้วยวิธีการเปรียบเทียบการประหยัด.....15
2.7 วิธีเมตาฮิวริสติก (Metaheuristic Method).....18
2.8 ตัวอย่างการใช้วิธีเมตาฮิวริสติกที่ใช้ในการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะ.....19
2.8.1 ขั้นตอนวิธีการค้นหาแบบทาบู (Tabu Search).....19
2.8.2 ขั้นตอนวิธีการจำลองการอบอ่อน (Simulated Annealing Algorithm).....20
2.9 การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ.....20
2.10 หลักการและทฤษฎีของภาษา VBA.....23

บทที่ 3 วิธีดำเนินโครงงาน.....25
3.1 ศึกษาปัญหาการจัดเส้นทางสำหรับยานพาหนะรูปแบบต่างๆ.....26
3.2 ศึกษารูปแบบปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ.....26
3.3 ศึกษาวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ.....26
3.4 เขียนโปรแกรมกระบวนการทำงานของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำลงบนคอมพิวเตอร์.....26
3.5 ตรวจสอบความถูกต้องของโปรแกรม.....26

สารบัญ (ต่อ)

หน้า

3.6 ใช้วิธีการหาคำตอบแบบวนรอบซ้ำเพื่อค้นหาคำตอบกับปัญหาทดสอบ.....27
3.7 สรุปผลและนำเสนอผลงาน.....27

บทที่ 4 ผลการทดลองและวิเคราะห์.....28
4.1 ออกแบบวิธีการค้นหาคำตอบเฉพาะที่สำหรับปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ.....28
4.1.1 การหาคำตอบเริ่มต้นที่ได้จากการสุ่ม.....28
4.1.2 การหาคำตอบใหม่ในย่านคำตอบเดิม.....29
4.1.3 กระบวนการจัดเรียงรอบรถ.....34
4.1.4 การออกแบบเมตาฮิวริสติก.....38
4.2 ความแตกต่างของ Local Search Type 1, Local Search Type 2 และ Iterated Local Search.....46
4.3 การใช้งานของโปรแกรม การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ.....47
4.3.1 กระบวนการจัดเรียงรอบรถ.....47
4.3.2 ทำการกรอกข้อมูลต่างๆ ลงในแบบฟอร์ม Data.....49
4.3.3 ตารางแสดงข้อมูลเบื้องต้นของลูกค้าแต่ละจุด.....50
4.3.4 ตารางแสดงผลลัพธ์ของการค้นหาคำตอบ.....52
4.3.5 ลักษณะการแจ้งข้อผิดพลาดที่เกิดขึ้นในการใช้โปรแกรม.....53
4.4 ผลการทดสอบโปรแกรม และวิเคราะห์ผลการทดลองในปัญหาขนาดเล็ก.....55
4.5 ผลการทดสอบโปรแกรม และวิเคราะห์ผลการทดลองในปัญหาขนาดกลาง.....59
4.6 ผลการทดสอบโปรแกรม และวิเคราะห์ผลการทดลองในปัญหาขนาดใหญ่.....63
4.7 การวิเคราะห์ผลการทดลอง.....67
4.7.1 การเปรียบเทียบผลจากหลักการในการทำ Neighborhood Search ทั้ง 3 หลักการกับระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะและเวลาในการประมวลผล.....67
4.7.2 ขนาดของปัญหากับระยะทางในการขนส่งสินค้าให้กับลูกค้าขของยานพาหนะและเวลาในการประมวลผล.....69
4.7.3 การวิเคราะห์ความแปรปรวน (ANOVA) ของผลลัพธ์โดยใช้โปรแกรม Minitab 16.....69

สารบัญ (ต่อ)

หน้า

4.7.4 การวิเคราะห์ข้อมูลโดยการทดสอบแบบ t-test....71
4.7.5 กระบวนการทดสอบสมมติฐาน....71
4.7.6 สรุปผลการวิเคราะห์ข้อมูลโดยการทดสอบแบบ t-test ของการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก....72

บทที่ 5 บทสรุปและข้อเสนอแนะ....72
5.1 บทสรุป....72
5.2 ปัญหาในการดำเนินโครงงานวิจัย....74
5.3 ข้อเสนอแนะ....74

เอกสารอ้างอิง....75

ภาคผนวก Source Code ของโปรแกรมการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก....76

ประวัติผู้ดำเนินโครงงาน....123

สารบัญตาราง

ตารางที่ หน้า

1.1 ขั้นตอนและแผนการดำเนินโครงงาน....3

2.1 คำสั่งซื้อ....15

2.2 แสดงระยะทางในการขนส่ง....15

2.3 การจัดเส้นทางการเดินรถ....17

2.4 ผลการจัดเส้นทาง ตรวจสอบคำตอบ 4 เส้นทาง....18

4.1 แสดงตัวอย่างความต้องการของลูกค้าทั้ง 6 ราย....35

4.2 ผลลัพธ์ของปัญหาที่ 1 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดเล็ก....56

4.3 ผลลัพธ์ของปัญหาที่ 2 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดเล็ก....58

4.4 ผลลัพธ์ของปัญหาที่ 1 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดกลาง....60

4.5 ผลลัพธ์ของปัญหาที่ 2 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดกลาง....61

4.6 ผลลัพธ์ของปัญหาที่ 1 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดใหญ่....63

4.7 ผลลัพธ์ของปัญหาที่ 2 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดใหญ่....65

4.8 แสดงผลลัพธ์จากการคำตอบที่ดีที่สุดของแต่ละวิธีการที่ใช้ในการค้นหาคำตอบที่ดีที่สุด....68

4.9 สรุปผลการวิเคราะห์ความแปรปรวน....70

4.10 สรุปผลการทดสอบ t-test....72

สารบัญรูป

รูปที่ หน้า

2.1 องค์ประกอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะ....5
2.2 การจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ....8
2.3 การจัดส่งสินค้าไป – กลับทุกๆ จุดส่งสินค้า....12
2.4 การรวมจุดส่งสินค้าเข้าด้วยกัน....12
2.5 ขั้นตอนการหาเส้นทางการขนส่งสินค้าด้วยวิธีการเปรียบเทียบการประหยัด....13
2.6 การปรับปรุงเส้นทางการเดินรถด้วยวิธีการเปรียบเทียบการประหยัด....16
2.7 การค้นหาพื้นที่คำตอบที่ดีที่สุดของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ....21
2.8 รหัสเทียมของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ....23
3.1 แสดงขั้นตอนการดำเนินงานวิจัยของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดด้านความจุ....25
4.1 ตัวอย่างคำตอบปัจจุบัน....29
4.2 การค้นหาคำตอบแบบสลับ 1 ของตำแหน่งหลักตำแหน่งที่ 1....30
4.3 การหาคำตอบแบบสลับ 1 ของตำแหน่งหลักที่ 2....30
4.4 การค้นหาคำตอบแบบสลับ 2 ของตำแหน่งหลักที่ 1....32
4.5 การค้นหาคำตอบแบบเลื่อน กรณีที่ตำแหน่งที่สุ่มมานั้นมีค่ามากกว่าตำแหน่งหลัก....33
4.6 การค้าหาคำตอบแบบเลื่อน กรณีที่ตำแหน่งที่สุ่มมานั้นมีค่าน้อยกว่าตำแหน่งหลัก....34
4.7 แสดงกระบวนการจัดเรียงรอบรถ....36
4.8 แสดงกระบวนการจัดเรียงรอบรถสินค้ามีปริมาณมากกว่าจำนวนรถ....37
4.9 การทำงานของ Local Search Type 1....38
4.10 แสดงการทำงานของ Local Search Type 2....41
4.11 แสดงการทำงานของ Iterated Local Search....44
4.12 หน้าแรกของโปรแกรม....47
4.13 User form Help ช่วยแนะนำวิธีการใช้โปรแกรม....48
4.14 User form Confirm Exit เพื่อสอบถามผู้ใช้งานว่าต้องการออกจากโปรแกรมหรือไม่....48
4.15 การรับข้อมูลต่างๆ ของโปรแกรม....49
4.16 แสดงผลข้อมูลที่ผู้ใช้กรอก....50
4.17 หน้าต่างให้ยืนยันการลบข้อมูลของปุ่ม Clear Data....51
4.18 หน้าต่างแจ้งเตือนเมื่อผู้ใช้กรอกข้อมูลเป็นตัวหนังสือ....51
4.19 แสดงผลลัพธ์ของการค้นหาคำตอบ....52
4.20 การแจ้งความผิดพลาดที่เกิดจากกรอกข้อมูลเป็นตัวหนังสือไม่ใช่ตัว....53

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.21 การแจ้งความผิดพลาดที่เกิดจาการกรอกค่าเป็นจุดทศนิยม....54
4.22 การแจ้งความผิดพลาดที่ผู้ใช้ลืมกรอกค่า....55
ก.1 หน้าแรกที่เป็นต้อนรับของโปรแกรมโดยอัตโนมัติ....77
ก.2 แสดงคำสั่งเมื่อคลิกปุ่ม START ในหน้าแรกของโปรแกรม....77
ก.3 แสดงคำสั่งเมื่อคลิก HELP ในหน้าแรกของโปรแกรม....78
ก.4 แสดงคำสั่งเมื่อคลิก EXIT ในหน้าแรกของโปรแกรม....78
ก.5 การทำงานของ User form Data....78
ก.6 การแจ้งเตือน....81
ก.7 การจัดกลางตัวหนังสือบน Excel....85
ก.8 แสดงการเลื่อน....86
ก.9 โค้ดการทำงานของ Worksheets Data....88
ก.10 การทำงานของปุ่ม Save....91
ก.11 การทำงานของปุ่ม Clear....92
ก.12 โค้ดของ User form Confirm Reset....92
ก.13 การทำงานของ Worksheets Run of Cycle....97
ก.14 Back 2....100
ก.15 Save 2....100
ก.16 การทำงานใน Module....102
ก.17 โค้ดการสร้างตาราง....102
ก.18 โค้ดการสุ่มหาคำตอบเริ่มต้น....103
ก.19 การคัดลอกชุดคำตอบเริ่มต้นเพื่อทำการปรับปรุงคำตอบ....104
ก.20 การปรับปรุงคำตอบแบบสลับ 1....105
ก.21 การปรับปรุงคำตอบแบบสลับ....106
ก.22 การปรับปรุงคำตอบแบบเลื่อน....108
ก.23 โค้ดการจัดลำดับของลูกค้าให้กับยานพาหนะโดยการพิจารณาความจุสินค้าที่ลูกค้าต้องการ....109
ก.24 โค้ดการคำนวณระยะทางในการเดินทางของยานพาหนะ....111
ก.25 การค้นหาคำตอบที่ดีที่สุดของ Local Search 1....115
ก.26 การค้นหาคำตอบที่ดีที่สุดของ Local Search 2....119
ก.27 การค้นหาคำตอบที่ดีที่สุดของ Iterated Local Search....123

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

ในช่วงเวลาที่ผ่านมาสภาพการแข่งขันทางเศรษฐกิจการค้าในประเทศได้มีการเปลี่ยนแปลงอย่างรวดเร็วอันเนื่องมาจากการเปิดเสรีทางการค้า และการรวมกลุ่มทางเศรษฐกิจในภูมิภาคต่างๆ ซึ่งในการเปลี่ยนแปลงดังกล่าว ทำให้มีอุตสาหกรรมที่หลากหลายเข้ามาลงทุนในประเทศไทยมากยิ่งขึ้น แม้ว่าประเทศไทยจะมีโครงสร้างพื้นฐานในด้านการขนส่งที่เพียงพอในระดับหนึ่ง เนื่องจากภาครัฐมีการลงทุนด้านโครงสร้างพื้นฐานอย่างต่อเนื่อง เช่น การวางโครงข่ายถนนทั่วประเทศ การสร้างสนามบินแห่งใหม่ เป็นต้น แต่อย่างไรก็ตาม การขนส่งยังกระจุกตัวอยู่ที่การใช้รถบรรทุกเป็นหลัก และในปัจจุบันเศรษฐกิจมีสภาวะราคาน้ำมันที่ปรับตัวสูงขึ้นอย่างต่อเนื่อง ส่งผลกระทบต่อภาคธุรกิจและอุตสาหกรรมต่างๆ เป็นอย่างมาก โดยเฉพาะอย่างยิ่งองค์กรที่ต้องอาศัยการขนส่งเป็นกิจกรรมหลักจะได้รับผลกระทบโดยตรง หากองค์กรเหล่านี้มีการจัดเส้นทางสำหรับยานพาหนะที่ไม่เหมาะสม จะทำให้ต้นทุนสูงขึ้น ส่งผลถึงศักยภาพในการแข่งขันลดลง ซึ่งปัญหาการจัดเส้นทางยานพาหนะเป็นการกำหนดเส้นทางของยานพาหนะแต่ละคันเพื่อไปให้บริการลูกค้า โดยเสียค่าใช้จ่ายในการขนส่งน้อยที่สุด หรืออีกนัยหนึ่งคือการพยายามจัดให้มีเส้นทางการเดินทางที่สั้นที่สุดเท่าที่จะเป็นไปได้นั่นเอง โดยพิจารณาถึงเงื่อนไขในการบรรจุสินค้าที่ขนส่ง ช่วยประหยัดค่าใช้จ่ายได้

สำหรับโครงงานวิจัยนี้ได้นำเสนอการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก ซึ่งปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ (Capacitated Vehicle Routing Problem : CVRP) คือ ปัญหาการตัดสินใจเลือกทางเลือกที่ดีที่สุดในการเคลื่อนย้ายสินค้า ภายใต้เงื่อนไขความสามารถในการบรรจุสินค้าของยานพาหนะ โดยการหาวิธีการวางแผนจัดลำดับและเส้นทางการขนส่งสินค้าไปยังลูกค้าที่มีประสิทธิภาพมากที่สุด เพื่อให้ลูกค้าเกิดความพึงพอใจในบริการ ซึ่งวัตถุประสงค์เพื่อให้มีระยะทางรวมในการเดินทางของยานพาหนะที่สั้นที่สุด ทำให้ลดต้นทุนในระบบการขนส่ง และได้เปรียบคู่แข่งทางการค้า และวิธีเมตาวิธีฮิวริสติกที่ใช้ในโครงงานวิจัยนี้ คือ การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ (Iterated Local Search : ILS)

1.2 วัตถุประสงค์ของโครงงาน

เพื่อสร้างวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ สำหรับปัญหาการจัดเส้นทางยานพาหนะภายใต้เงื่อนไขความสามารถในการบรรจุสินค้า ซึ่งจะต้องมีระยะทางในการขนส่งที่สั้นที่สุด

1.3 เกณฑ์การชี้วัดผลงาน (Output)

วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำและโปรแกรมที่สร้างขึ้นด้วยภาษา VBA (Visual Basic for Applications) บน Microsoft Excel

1.4 เกณฑ์ชี้วัดผลสำเร็จ (Outcome)

วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำและโปรแกรมที่สร้างขึ้นสามารถนำไปช่วยในการแก้ปัญหาการจัดเส้นทางยานพาหนะในการขนส่งแบบมีข้อจำกัดด้านความสามารถในการบรรจุสินค้าภายใต้เงื่อนไขที่กำหนด

1.5 ขอบเขตในการดำเนินโครงการ

1.5.1 ศึกษาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ โดยใช้วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ เพื่อให้ระยะทางรวมต่ำสุด ช่วยลดต้นทุนในระบบการขนส่งและกำหนดข้อจำกัดด้านความสามารถในการบรรจุสินค้า

1.5.2 ศึกษาแบบจำลองทางคณิตศาสตร์ของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ โดยมีข้อตกลงเบื้องต้นดังนี้

1.5.2.1 ยานพาหนะจะต้องเริ่มต้นและสิ้นสุด ณ ที่คลังสินค้า

1.5.2.2 ลูกค้าแต่ละรายจะได้รับสินค้าจากยานพาหนะเพียงคันเดียวเท่านั้น

1.5.2.3 ยานพาหนะแต่ละคันจะใช้เส้นทางใดเส้นทางหนึ่งเท่านั้น ในการขนส่งสินค้าให้ลูกค้า

1.5.2.4 ยานพาหนะสามารถขนส่งสินค้าให้กับลูกค้าได้ไม่เกินความจุที่ยานพาหนะสามารถบรรทุกได้

1.5.2.5 ยานพาหนะทุกคันมีความจุเท่ากันหมด

1.5.2.6 ความต้องการของลูกค้าแต่ละคนทราบค่าแน่นอนไม่เปลี่ยนแปลง

1.5.2.7 ระยะทางในการขนส่งสินค้าให้กับลูกค้าจุดหนึ่งไปยังลูกค้าอีกจุดหนึ่งทราบค่าแน่นอนไม่เปลี่ยนแปลง

1.5.2.8 วัตถุประสงค์ของปัญหาต้องการหาระยะทางรวม

1.5.3 การทดสอบประสิทธิภาพของเมตาฮิวริสติกจะทำการทดลองกับโจทย์ปัญหาที่ผู้จัดทำโครงงานสร้างขึ้น

1.6 สถานที่ในการดำเนินงาน

ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

1.7 ขั้นตอนและแผนการดำเนินโครงการ

มิถุนายน พ.ศ. 2555 ถึง มกราคม พ.ศ. 2556

1.8 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1 ขั้นตอนและแผนการดำเนินโครงงาน

ลำดับ	การดำเนินงาน	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.
1.8.1	กำหนดปัญหาที่ต้องการศึกษา	←	—	→					
1.8.2	ศึกษาปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ		←	—	→				
1.8.3	ศึกษาวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ		←	—	→				
1.8.4	ศึกษาการเขียนโปรแกรม				←	—	→		
1.8.5	การเขียนโปรแกรม					←	—	→	
1.8.6	ทำการทดสอบโปรแกรม และปรับปรุงแก้ไขโปรแกรม						←	—	→
1.8.7	พิมพ์รายงานและสรุปโครงงานวิจัย								← →

บทที่ 2
หลักการและทฤษฎีเบื้องต้น

2.1 ปัญหาการจัดเส้นทางสำหรับยานพาหนะ (Vehicle Routing Problem : VRP)

ปัญหาการจัดเส้นทางสำหรับยานพาหนะ เป็นปัญหาด้านการขนส่งและลอจิสติกส์รูปแบบหนึ่ง ที่มีการศึกษามายาวนานกว่า 40 ปี และการค้นคว้าอย่างแพร่หลาย โดยมีการเพิ่มเงื่อนไขและข้อจำกัดต่างๆ เพื่อให้เหมาะสมกับสถานการณ์ที่สามารถนำไปใช้งานจริงได้มากขึ้น ทำให้ปัญหาการจัดเส้นทางสำหรับยานพาหนะ ได้รับความนิยมและมีการพัฒนาจนมีความหลากหลายมากขึ้นตามไปด้วย ซึ่งปัญหาการจัดเส้นทางสำหรับยานพาหนะนี้ คือ ปัญหาในการตัดสินใจเลือกทางเลือกที่ดีที่สุดในการขนส่งสินค้า โดยการหาวิธีการวางแผนจัดลำดับและเส้นทางการขนส่งสินค้าไปยังลูกค้าหรือผู้บริโภคที่มีประสิทธิภาพมากที่สุด เพื่อให้ลูกค้าเกิดความพึงพอใจในบริการ เป้าหมายสำคัญของปัญหาการจัดเส้นทางยานพาหนะ คือ การพยายามออกแบบกลุ่มของยานพาหนะทุกคัน ให้เหมาะสมกับข้อกำหนดต่างๆ เพื่อให้เสียค่าใช้จ่ายในการขนส่งต่ำที่สุด โดยการลดจำนวนรถขนส่งสินค้าหรือเพื่อลดค่าใช้จ่ายต้นทุนคงที่ (Fixed Cost) ที่เกิดขึ้นในการขนส่งแต่ละครั้ง เมื่อจำนวนรถลดลง ความจำเป็นในการจ้างพนักงานขับรถเพิ่มจึงน้อยลงตามไปด้วย รวมทั้งการลดระยะทางในการเดินทางหรือลดระยะเวลาในการเดินทาง เมื่อระยะทางและระยะเวลาลดลง ค่าใช้จ่ายต้นทุนแปรผัน (Variable Cost) ที่เกิดขึ้นในการขนส่งแต่ละครั้งจะลดลงตามไปด้วยค่าใช้จ่ายต้นทุนแปรผัน ได้แก่ ค่าน้ำมันและค่าใช้จ่ายต่างๆ ที่เกิดขึ้นบนเส้นทางนั้นๆ ซึ่งมีจุดเริ่มต้นและสิ้นสุดที่คลังสินค้า (Depot) โดยยานพาหนะวิ่งไปตามเส้นทางที่จะส่งสินค้า ซึ่งจะพิจารณาถึงเงื่อนไขหรือข้อจำกัดต่างๆ ด้วย เช่น เวลา, จำนวนยานพาหนะ และระยะทาง เป็นต้น

2.2 ปัญหาการจัดเส้นทางยานพาหนะ

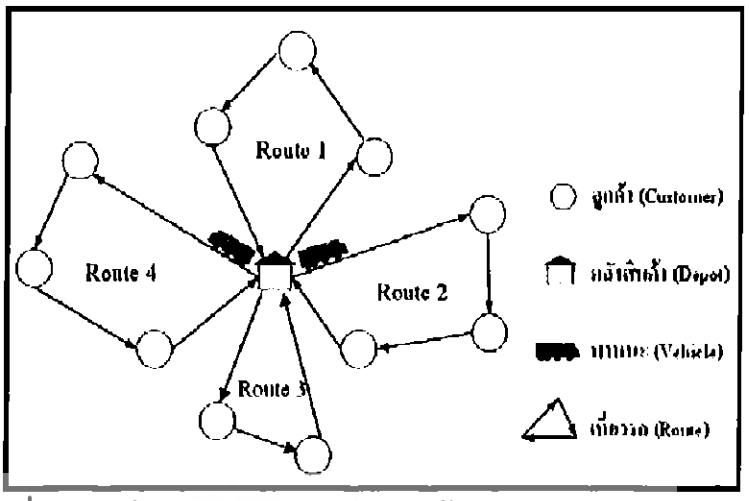
โดยทั่วไปแล้วปัญหาการจัดเส้นทางยานพาหนะนั้น จะประกอบไปด้วย

2.2.1 กลุ่มลูกค้า (Set of Customer) ลูกค้าแต่ละรายจะถูกกำหนดให้อยู่กระจายในจุด (Node) ต่างๆ กัน และมีความต้องการรับหรือส่งสินค้าในจำนวนต่างๆ กัน

2.2.2 ยานพาหนะ (Vehicle) หมายถึง รถ เรือ เครื่องบิน หรือยานพาหนะอื่นๆ ที่ใช้ในการบริการให้แก่ลูกค้า มีหน้าที่ในการเดินทางรับ - ส่งสินค้าระหว่างลูกค้าและคลังสินค้า

2.2.3 คลังสินค้า (Depot) หมายถึง สถานที่เก็บสินค้า โรงงาน หรือศูนย์กระจายสินค้า เป็นสถานที่ที่กำหนดให้เป็นจุดเริ่มต้นและจุดสิ้นสุดในการเดินทาง

2.2.4 เที่ยวรถ (Route) คือ เส้นทางการเดินรถที่มีการกำหนดว่าจะให้พาหนะคันใดเดินทางไปยังลูกค้ารายใดบ้าง และเดินทางตามลำดับก่อนหลังอย่างไร แสดงดังรูปที่ 2.1



รูปที่ 2.1 องค์ประกอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะ

ที่มา : ธารชุดา พันธ์นิกุล, 2551

จากรูปที่ 2.1 องค์ประกอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะ ซึ่งจะเห็นได้ว่า ยานพาหนะแต่ละคันจะเริ่มต้นการเดินทาง ณ ที่คลังสินค้า (Depot) โดยจะวิ่งไปตามเส้นทางที่จะส่งสินค้าให้กับลูกค้าแต่ละจุด ภายใต้เงื่อนไขและข้อจำกัดต่างๆ ซึ่งจะเป็นเส้นทางการขนส่งที่มีระยะทางที่สั้นที่สุด และเมื่อยานพาหนะเดินทางส่งสินค้าให้กับลูกค้าครบทุกจุดตามที่กำหนดไว้แล้วก็จะเดินทางกลับมาที่คลังสินค้าเดิม โดยในรูปมีเที่ยวรถอยู่ 4 เส้นทาง

2.3 รูปแบบของปัญหาการจัดเส้นทางสำหรับยานพาหนะ

จากการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง พบว่างานวิจัยที่เกี่ยวข้องกับปัญหาการจัดเส้นทางยานพาหนะ ได้รับความนิยมอย่างแพร่หลายทั้งในและต่างประเทศ Dantzing and Ramser (1959) ได้นำเสนอวิธีการแก้ปัญหาการจัดเส้นทางยานพาหนะ ซึ่งนับเป็นจุดเริ่มต้นของการพัฒนาขั้นตอนวิธีการต่างๆ ในการแก้ปัญหาดังกล่าว มาจนถึงกระทั่งปัจจุบันนี้ ซึ่งมีทั้งการพัฒนาที่รูปแบบของปัญหา และเทคนิควิธีการแก้ปัญหา ดังแสดงในงานวิจัยของ Toth and Vigo (2001) ได้แบ่งปัญหาการจัดเส้นทางยานพาหนะไว้ ดังต่อไปนี้

2.3.1 ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ (Capacitated Vehicle Routing Problem : CVRP)

ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ จะเป็นการจัดเส้นทางยานพาหนะที่กำหนดให้ยานพาหนะทุกคันต้องออกจากคลังสินค้า และกลับมายังคลังสินค้าเดิมทุกครั้งเพื่อขนส่งสินค้าให้ครบตามปริมาณความต้องการในแต่ละบัพ (บัพ คือ ตำแหน่งลูกค้าเป้าหมายในการขนส่ง) หรือจุดยอด ซึ่งยานพาหนะจะต้องบรรทุกสินค้ารวมทั้งหมดไม่เกินน้ำหนักบรรจุที่บรรจุได้สูงสุด ยกตัวอย่างเช่น โรงงานน้ำดื่มแห่งหนึ่งทางโรงงาน มีคลังสินค้าเพียงแห่งเดียว

และมีรถบรรทุกสินค้า 1 คันที่ใช้ไปส่งสินค้าไปยังลูกค้ารายต่างๆ รถบรรทุกที่ใช้ในการขนส่งสินค้ามีความจุในการบรรทุกได้สูงสุดไม่เกิน 2,100 กิโลกรัม ดังนั้นทางโรงงานจึงต้องวางแผนลำดับในการส่งสินค้าให้กับลูกค้าก่อน - หลัง ภายใต้เงื่อนไขความจุของรถบรรทุกที่บรรทุกได้สูงสุดไม่เกิน 2,100 กิโลกรัม ซึ่งจะต้องมีระยะทางในการขนส่งที่สั้นที่สุด เป็นต้น

2.3.2 ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบหลายคลังสินค้า (Multiple - Depots Vehicle Routing Problem : MDVRP)

ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบหลายคลังสินค้า เป็นปัญหาการขนส่งอีกรูปแบบหนึ่งที่กำลังเป็นที่สนใจของระบบลอจิสติกส์ เนื่องจากการกระจายในปัจจุบันนั้น มักจะเป็นการกระจายสินค้าผ่านคลังสินค้าที่มีมากกว่า 1 แห่ง ซึ่งจะมีวิธีการแก้ปัญหาคล้ายคลึงกับปัญหาการจัดเส้นทางยานพาหนะ แต่เมื่อประกอบด้วยหลายคลังสินค้า จึงทำให้จำนวนตัวแปรและเงื่อนไขข้อจำกัดมีจำนวนเพิ่มมากขึ้นเป็นเท่าตัวตามจำนวนคลังสินค้า ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบหลายคลังสินค้า จึงค่อนข้างมีความยุ่งยากและซับซ้อนมากกว่าปัญหาการจัดเส้นทางยานพาหนะ ยกตัวอย่างเช่น NT transport จำกัด เป็นบริษัทที่รับผิดชอบการขนส่งเงินสดระหว่างศูนย์กระจายเงินสด 3 แห่ง (ศูนย์ A, ศูนย์ B และศูนย์ C) ไปยังสาขาต่างๆ ที่ตั้งอยู่ในเขตกรุงเทพฯ และปริมณฑล การแบ่งจำนวนลูกค้า (สาขา) ของแต่ละศูนย์กระจายเงินสดนั้น แบ่งตามความสามารถของกระบวนการจัดเตรียมเงินสดของแต่ละศูนย์เป็นสำคัญ โดยความสามารถในการจัดการจัดเตรียมเงินสดของแต่ละศูนย์นั้นไม่เท่ากัน ซึ่งสัดส่วนความสามารถในการจัดเตรียมของศูนย์ A, ศูนย์ B และศูนย์ C เท่ากับ ร้อยละ 50, ร้อยละ 30 และร้อยละ 20 ตามลำดับ

2.3.3 ปัญหาการจัดเส้นทางสำหรับพาหนะแบบมีข้อจำกัดเนื่องจากเวลา (Vehicle Routing Problem with Time Windows : VRPTW)

สำหรับปัญหาการจัดเส้นทางสำหรับพาหนะแบบมีข้อจำกัดเนื่องจากเวลา เป็นรูปแบบปัญหาการจัดเส้นทางยานพาหนะที่เพิ่มเงื่อนไขเกี่ยวกับระยะเวลาที่ลูกค้ากำหนดให้มีการส่งสินค้าภายในกรอบเวลา (Time Window) โดยการจัดลูกค้าเข้าสู่เส้นทางการเดินรถ เพื่อให้เกิดต้นทุนต่ำสุดหรือเวลาเดินทางสั้นที่สุด ซึ่งปัญหาลักษณะนี้จะเพิ่มความยุ่งยากและซับซ้อนมากยิ่งขึ้น ถึงแม้จะสร้างขั้นตอนวิธีทางคอมพิวเตอร์เพื่อใช้ในการหาผลเฉลย อาจจะต้องประสบปัญหาการใช้เวลาในการค้นหาผลเฉลย เมื่อลูกค้าและเงื่อนไขของปัญหามีเป็นจำนวนมาก เวลาที่ใช้ในการหาผลเฉลยก็จะเพิ่มขึ้นแบบเลขชี้กำลัง (Exponential)

2.3.4 ปัญหาการจัดเส้นทางสำหรับพาหนะแบบมีข้อจำกัดเนื่องจากความไม่แน่นอน (Stochastic Vehicle Routing Problem : SVRP)

ปัญหาการจัดเส้นทางสำหรับพาหนะแบบมีข้อจำกัดเนื่องจากความไม่แน่นอน ปรับปรุงมาจากปัญหาการจัดเส้นทางยานพาหนะ เพื่อทำให้ปัญหามีความเสมือนจริงมากขึ้น เนื่องจากปัจจุบันเงื่อนไขต่างๆ มีความไม่แน่นอน เช่น ความต้องการของลูกค้า ระยะทางที่ใช้ในการเดินทาง ระยะเวลาในการให้บริการ เป็นต้น ซึ่งความไม่แน่นอนของปัญหาสามารถแบ่งเป็น 2 แบบ ดังนี้

2.3.4.1 ปัญหาการจัดเส้นทางยานพาหนะแบบมีลูกค้าไม่แน่นอน (VRP with Stochastic Customer : VRPSC) และปัญหาการจัดเส้นทางยานพาหนะแบบมีความต้องการไม่แน่นอน (VRP with Stochastic Demands : VRPSD) ซึ่งเป็นปัญหาที่ลูกค้าและความต้องการไม่แน่นอน มีงานวิจัยเกิดขึ้นในส่วนนี้อย่างกว้างขวาง

2.3.4.2 ปัญหาการจัดเส้นทางยานพาหนะแบบมีระยะเวลาเดินทางไม่แน่นอน (VRP with Stochastic Travel Time : VRPSTT) และปัญหาการจัดเส้นทางยานพาหนะ แบบมีระยะเวลาในการให้บริการไม่แน่นอน (VRP with Stochastic Service Time : VRPSST) ซึ่งปัญหาในรูปแบบนี้เมื่อเทียบกับปัญหาการจัดเส้นทางยานพาหนะแบบมีลูกค้าไม่แน่นอน และปัญหาการจัดเส้นทางยานพาหนะแบบมีความต้องการไม่แน่นอนแล้ว ค่อนข้างได้รับความสนใจน้อยกว่า ซึ่งอาจเป็นได้ว่าปัญหาดังกล่าวอาจจะใช้งานได้ยากในกรณีที่ความไม่แน่นอนของสภาวะแวดล้อมของเส้นทางเดินรถและสภาพจราจรในท้องถนนที่มีการจราจรค่อนข้างหนาแน่น ซึ่งเป็นปัจจัยที่ค่อนข้างควบคุมได้ยากสำหรับพื้นที่ในกรุงเทพฯ และปริมณฑล เพราะไม่สามารถคาดการณ์เวลาที่แน่นอนได้เลย

2.3.5 ปัญหาการจัดเส้นทางสำหรับพาหนะแบบมีการรับและส่งสินค้า (Vehicle Routing Problem with Pick – Up and Delivery : VRPPD)

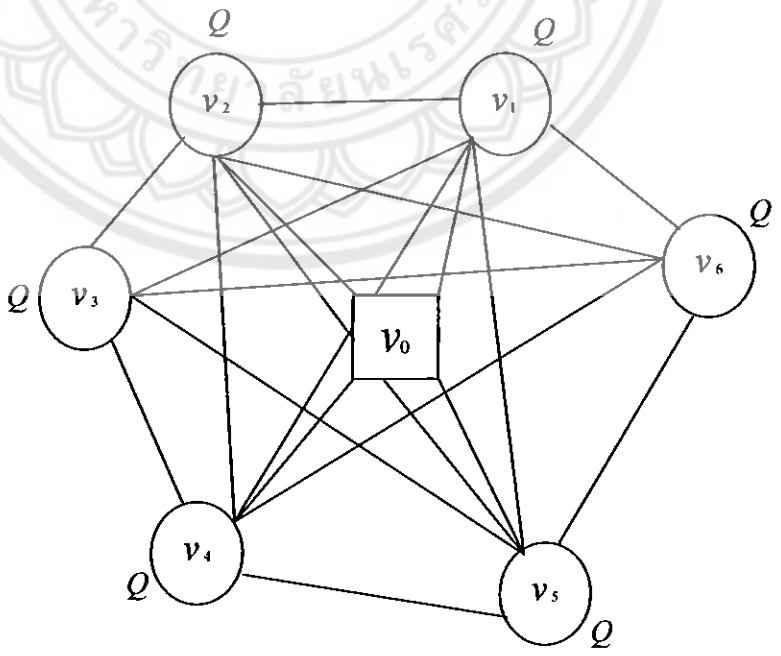
ปัญหาการจัดเส้นทางสำหรับพาหนะแบบมีการรับและส่งสินค้า ปัญหาในลักษณะนี้จะมีความยุ่งยากเพิ่มขึ้นเพราะมีข้อจำกัดเกี่ยวกับลำดับของลูกค้าที่ต้องเดินทางไปก่อนหน้า ทั้งยังมีข้อจำกัดเนื่องจากความจุ (Capacity) ของพาหนะที่ใช้ในการขนส่ง และต้องมีการรับและการส่งมอบสินค้า ยกตัวอย่างความยุ่งยากของปัญหา เช่น หากพาหนะบรรทุกสินค้าจนเต็มความจุแล้วก็จะไม่สามารถรับสินค้าเพิ่มได้อีก แต่ต้องเดินทางไปหาลูกค้าที่ต้องการส่งมอบ (Delivery Customer) ก่อนเพื่อระบายสินค้าออกไปจึงจะรับสินค้าเพิ่มได้ เป็นต้น

2.4 ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ

ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ จะเป็นการจัดเส้นทางยานพาหนะที่กำหนดให้ยานพาหนะทุกคันต้องเริ่มต้น และสิ้นสุดที่คลังสินค้าเดิมทุกครั้ง ซึ่งยานพาหนะจะวิ่งไปตามเส้นทางที่จะส่งสินค้า โดยจะพิจารณาถึงเงื่อนไขหรือข้อจำกัดด้านความจุของ

ยานพาหนะ กล่าวคือ ยานพาหนะจะต้องบรรทุกสินค้ารวมทั้งหมดไม่เกินน้ำหนักความจุที่บรรจุได้สูงสุด

แบบจำลองทางคณิตศาสตร์ (Mathematical Model) ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ สามารถอธิบายได้จากกราฟของ $G=(V, A)$ เมื่อ $V=\{v_0, v_1, v_2, \ldots, v_n\}$ คือ เซตของโนดหรือตำแหน่งของลูกค้า โดยที่โนด v_0 คือ คลังสินค้า ส่วน $v_1, v_2, \ldots, v_n$ คือ ตำแหน่งของลูกค้ารายที่ $1, 2, \ldots, n$ และ $A=\{(v_i, v_j): i \neq j\}$ คือ เซตของเส้นเชื่อมระหว่างตำแหน่ง i ไปยังตำแหน่ง j หรือที่เรียกว่า อาร์ค (Arc) โดย $D=\{d_{ij}, i \neq j\}$ คือ ระยะทางระหว่างตำแหน่ง i ไปยังตำแหน่ง j โดยที่ค่าของ (v_i, v_j) และ d_{ij} จะต้องไม่เป็นลบ ซึ่งค่า d_{ij} จะสอดคล้องกับระยะห่างระหว่าง v_i และ v_j ในด้านต้นทุนและเวลาระหว่างทั้งสองตำแหน่ง และ q_i คือ ความต้องการสินค้าของลูกค้ารายที่ i ซึ่งในกรณีดังกล่าวนี้จะมีวัตถุประสงค์ คือ เพื่อลดต้นทุนรวมในการขนส่งสินค้าและในขณะเดียวกันจะต้องเป็นไปตามเงื่อนไขดังนี้ คือ (1) ลูกค้าทุกคนจะได้รับสินค้าจากการส่งสินค้าของยานพาหนะเพียงครั้งเดียว (2) ยานพาหนะทุกคันจะเริ่มต้นและสิ้นสุดที่คลังสินค้า (3) ความต้องการสินค้าของลูกค้ารวมทั้งหมดจะต้องไม่เกินความจุที่ยานพาหนะแต่ละคันสามารถบรรทุกได้ และเพื่อเพิ่มความเข้าใจยิ่งขึ้นสามารถดูได้จากรูปประกอบการบรรยายรูปที่ 2.2 แสดงจุดเชื่อมต่อจากคลังสินค้าไปยังลูกค้าแต่ละตำแหน่งและระหว่างลูกค้าตำแหน่งหนึ่งไปยังลูกค้าอีกตำแหน่งหนึ่ง โดยยานพาหนะแต่ละคันมีความจุเท่ากับ Q



รูปที่ 2.2 แสดงจุดเชื่อมต่อจากคลังสินค้าและลูกค้าแต่ละตำแหน่ง

ซึ่งปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ สามารถเขียนเป็นแบบจำลองกำหนดการเชิงคณิตศาสตร์ได้ ดังนี้

ดัชนี

i, j = ใช้แทนลำดับของคลังสินค้าและลูกค้า โดยกำหนดให้คลังสินค้ามี $i, j = 0$ ลูกค้าอื่นๆ จะมี $i, j = 1, 2, \dots, n$

k = ใช้แทนพาหนะคันที่ $1, 2, \dots, k$

ตัวแปร

v = $\{v_0, v_1, v_2, \dots, v_n\}$ เมื่อ v_0 = คลังสินค้า และอื่นๆ เป็นลูกค้าแต่ละจุด

n = จำนวนลูกค้าและคลังสินค้าทั้งหมด

d_{ij} = ระยะทางระหว่างลูกค้า หรือคลังสินค้าตำแหน่งที่ i ไปยัง j

q_i = ความต้องการสินค้าของลูกค้าลำดับที่ i เมื่อ $q_0 = 0$

Q = ความสามารถในการบรรทุกสินค้าของยานพาหนะ

K = $\{k_1, k_2, \dots, k_m\}$

m = จำนวนยานพาหนะทั้งหมด

x_{ij}^k = 1 ถ้ายานพาหนะ k ขนส่งสินค้าระหว่างลูกค้า i ไปยังลูกค้า j

= 0 ในกรณีอื่นๆ

Objective Function

$$Min \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}^k \tag{2.1}$$

s.t.

$$\sum_{k \in V} \sum_{j \in N} x_{ij}^k = 1 \tag{2.2}$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{ji}^k = 0, \forall i \in V, k \in K \tag{2.3}$$

$$\sum_{j \in V} x_{0j}^k = 1, \forall k \in K \tag{2.4}$$

$$\sum_{j \in V} x_{j0}^k = 1, \forall k \in K \tag{2.5}$$

$$\sum_{j \in V} q_j \left(\sum_{i \in V} x_{ij}^k \right) \leq Q, \forall k \in K \quad (2.6)$$

$$x_{ij}^k \in \{0,1\}, \forall i, j \in V, k \in K \quad (2.7)$$

Objective Function 2.1 คือ การพยายามหาเส้นทางการเดินทางของยานพาหนะที่ทำให้ผลรวมของระยะทางที่ยานพาหนะทุกคันวิ่งไปยังลูกค้าทุกจุดมีค่าที่น้อยที่สุด s.t. ที่ 2.2 คือ การประกันว่าลูกค้าแต่ละรายจะได้รับสินค้าจากยานพาหนะเพียงคันเดียว ซึ่งหมายความว่าจะมียานพาหนะเดินทางไปยังลูกค้าแต่ละรายได้เพียงคันเดียวเท่านั้น s.t. ที่ 2.3 ควบคุมให้ยานพาหนะที่วิ่งออกจากคลังสินค้าจะต้องเท่ากับยานพาหนะที่วิ่งกลับเข้ามายังคลังสินค้าเดิม s.t. ที่ 2.4 และ 2.5 ประกันว่ายานพาหนะขนส่งแต่ละคันถูกใช้เพียงเส้นทางใดเส้นทางหนึ่งเท่านั้น s.t. ที่ 2.6 กำหนดให้ยานพาหนะขนส่งสินค้าทุกคันสามารถบรรทุกสินค้าได้ไม่เกินความจุของยานพาหนะที่สามารถบรรทุกได้ s.t ที่ 2.7 นิยามให้สมาชิกในเซตของตัวแปรสำหรับการตัดสินใจมีค่าเป็นไปได้เฉพาะ 0 หรือ 1 เท่านั้น คือ เกิดการเชื่อมโยงกัน หรือไม่ได้เชื่อมโยงกัน (ธารชุดา พันธ์นิกุล, 2551)

2.5 วิธีแก้ปัญหาในการหาค่าคำตอบที่ดีที่สุดของปัญหาการจัดเส้นทางสำหรับยานพาหนะ

ปัญหาการจัดเส้นทางสำหรับยานพาหนะ จัดเป็นปัญหาการตัดสินใจที่มีความซับซ้อนในระดับเอ็นพีฮาร์ด (NP - Hard Problem) ซึ่งวิธีการแก้ปัญหาเพื่อให้ได้คำตอบที่ดีที่สุดภายในเวลาที่เหมาะสมนั้นกระทำได้ยากหรือไม่สามารถทำได้เลย โดยเฉพาะเมื่อปัญหาที่มีขนาดใหญ่ และมีเงื่อนไขเพิ่มมากขึ้น เวลาที่ใช้ในการแก้ปัญหาก็จะเพิ่มขึ้นเป็นแบบเลขชี้กำลัง (Exponential) ซึ่งวิธีในการหาคำตอบที่ดีที่สุดนั้นจะมีวิธีหลักๆ อยู่ 2 วิธี คือ วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์ และวิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณค่า

2.5.1 วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์

วิธีการแก้ปัญหาในรูปแบบนี้จะอยู่บนพื้นฐานของหลักการทางคณิตศาสตร์เป็นหลัก ซึ่งจะมีขั้นตอน ตัวแปร และสมการที่ซับซ้อน และเนื่องจากรูปแบบปัญหาในปัจจุบันค่อนข้างที่จะมีความยุ่งยากและซับซ้อนมากขึ้น จึงทำให้วิธีการหาคำตอบที่ดีที่สุดโดยวิธีนี้นั้นเป็นไปได้ยาก ซึ่งวิธีนี้ไม่เหมาะสมกับปัญหาที่มีขนาดใหญ่ เพราะวิธีนี้ค่อนข้างที่จะยุ่งยากซับซ้อนและใช้เวลามากในการประมวลผลคำตอบ (มาลินี หลวงคลัง และคณะ, 2554)

2.5.2 วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณค่า

เนื่องจากวิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์เพื่อให้ได้คำตอบที่เหมาะนั้นค่อนข้างที่จะเป็นไปได้ยาก รวมทั้งใช้เวลาในการประมวลผลคำตอบนาน และเนื่องจากความล่าช้าในการหาผลเฉลยของปัญหา จึงได้มีการพัฒนาวิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณค่ามาใช้เพื่อให้สามารถค้นหาคำตอบได้รวดเร็วยิ่งขึ้นคำตอบ โดยจะค่อยๆ ทำการสร้างผลเฉลยขึ้นมาทีละส่วน จนกระทั่งผลเฉลยนั้นสมบูรณ์ ลักษณะการทำงานจะทำการวนซ้ำของรอบการทำงาน (Relation) ในการค้นหาคำตอบ และจะหยุดทำงานเมื่อถึงเงื่อนไขที่ได้กำหนดไว้ (มาลินี หลวงคลัง และคณะ, 2554)

2.6 วิธีฮิวริสติก (Heuristic Method)

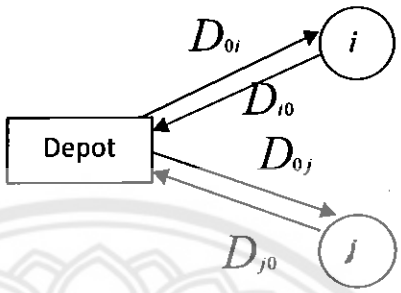
ฮิวริสติกมักจะถูกใช้ในการแก้ปัญหาที่หาคำตอบได้ยาก และสามารถใช้ในการหาทางแก้ปัญหาที่น่าพึงพอใจสำหรับปัญหาที่ซับซ้อน คือ เป็นวิธีการหาผลเฉลยที่อาจจะไม่ได้คำตอบที่ดีที่สุด แต่คุณภาพคำตอบที่ได้ก็ดีเพียงพอต่อความต้องการในการวางแผนต่างๆ ซึ่งวิธีฮิวริสติกถูกสร้างขึ้นเพื่อใช้ในการหาผลเฉลยของแต่ละปัญหาเท่านั้น ดังนั้นวิธีฮิวริสติกที่สามารถหาผลเฉลยที่ดีสำหรับปัญหาหนึ่งจึงไม่สามารถนำไปใช้หาผลเฉลยของอีกปัญหาหนึ่งได้

ซึ่งในกรณีปัญหาการจัดเส้นทางยานพาหนะสามารถสร้างแบบจำลองทางคณิตศาสตร์ได้ แต่ด้วยรูปแบบที่ยุ่งยากซับซ้อน รวมทั้งมีตัวแปรและเงื่อนไขในการตัดสินใจที่เพิ่มขึ้นมากไปด้วย การประยุกต์ใช้วิธีแม่นตรงก็คือ วิธีการคำนวณทุกทางเลือกของผลเฉลย แล้วเลือกผลเฉลยที่ให้ค่าที่ดีที่สุด หรือเทคนิควิธีการกำหนดการเชิงเส้น (Linear Programming) จึงอาจกระทำได้ยากหรือไม่สามารถกระทำได้เลย วิธีฮิวริสติกจึงน่าจะเป็นวิธีการที่เหมาะสมสำหรับปัญหาการจัดเส้นทางยานพาหนะโดยวิธีฮิวริสติกที่นิยมนำมาใช้แก้ปัญหาการจัดเส้นทางยานพาหนะ เช่น วิธีการเปรียบเทียบการประหยัด, 2 – Opt เป็นต้น ซึ่งวิธีฮิวริสติกถือว่าเป็นวิธีการที่สามารถใช้ในการแก้ปัญหาการจัดเส้นทางยานพาหนะได้อย่างมีประสิทธิภาพ ซึ่งในที่นี้จะกล่าววิธีฮิวริสติกวิธีหนึ่งที่ได้รับความนิยมในการนำมาใช้แก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ ดังนี้

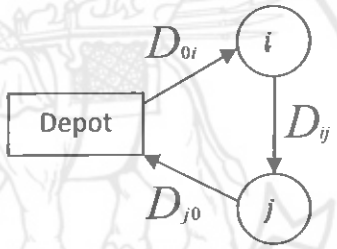
2.6.1 การจัดเส้นทางสำหรับยานพาหนะแบบมีเงื่อนไขเนื่องจากความจุ ด้วยวิธีการเปรียบเทียบการประหยัด (Saving Algorithm)

วิธีการเปรียบเทียบการประหยัด ถูกพัฒนาขึ้นโดย Clarke and Wright (1964) สำหรับแก้ปัญหาการจัดเส้นทางยานพาหนะ โดยเฉพาะวิธีการนี้เป็นการสร้างผลเฉลยทีละขั้นตอน โดยเริ่มจากผลเฉลยเริ่มต้นซึ่งอาจยังมีความเป็นไปไม่ได้ และสร้างผลเฉลยในลำดับถัดมาที่ทำให้ฟังก์ชันของการเปรียบเทียบการประหยัดมีค่าเพิ่มมากขึ้น หรือเลือกแทรกลูกค้าเข้ามาในเส้นทางเดิมที่มีอยู่ โดยที่ความต้องการรวมต้องไม่เกินความสามารถในการบรรทุกสินค้าของยานพาหนะที่ใช้ในการขนส่ง การแทรกจะเกิดขึ้นเรื่อยๆ จนกระทั่งได้เส้นทางที่เหมาะสมที่สุด โดยขั้นตอนมีดังนี้

ขั้นตอนที่ 1 คำนวณหาค่าความประหยัดของจุดส่งสินค้าแต่ละคู่จากการรวมจุดส่งสินค้าต่างๆ เข้าไว้ในเส้นทางหลัก ดังรูปที่ 2.3 แทนการจัดส่งสินค้าจากคลังสินค้า (Depot) ไปและกลับยังทุกๆ จุดส่งสินค้า ดังรูปที่ 2.4 ทำให้เกิดความประหยัดในการเดินทางจะได้ค่าความประหยัด คือ $\left(D_{0i} + D_{i0} + D_{0j} + D_{j0} \right) - \left(D_{0i} + D_{j0} + D_{ij} \right) = D_{i0} + D_{0j} + D_{ij}$



รูปที่ 2.3 การจัดส่งสินค้าไป – กลับทุกๆจุดส่งสินค้า



รูปที่ 2.4 การรวมจุดส่งสินค้าเข้าด้วยกัน

จะได้สมการในการคำนวณหาค่าความประหยัด คือ $S_{ij} = D_{i0} + D_{0j} + D_{ij}$

เมื่อ S_{ij} = ค่าความประหยัดระหว่างคู่จุดส่งสินค้า i และ j

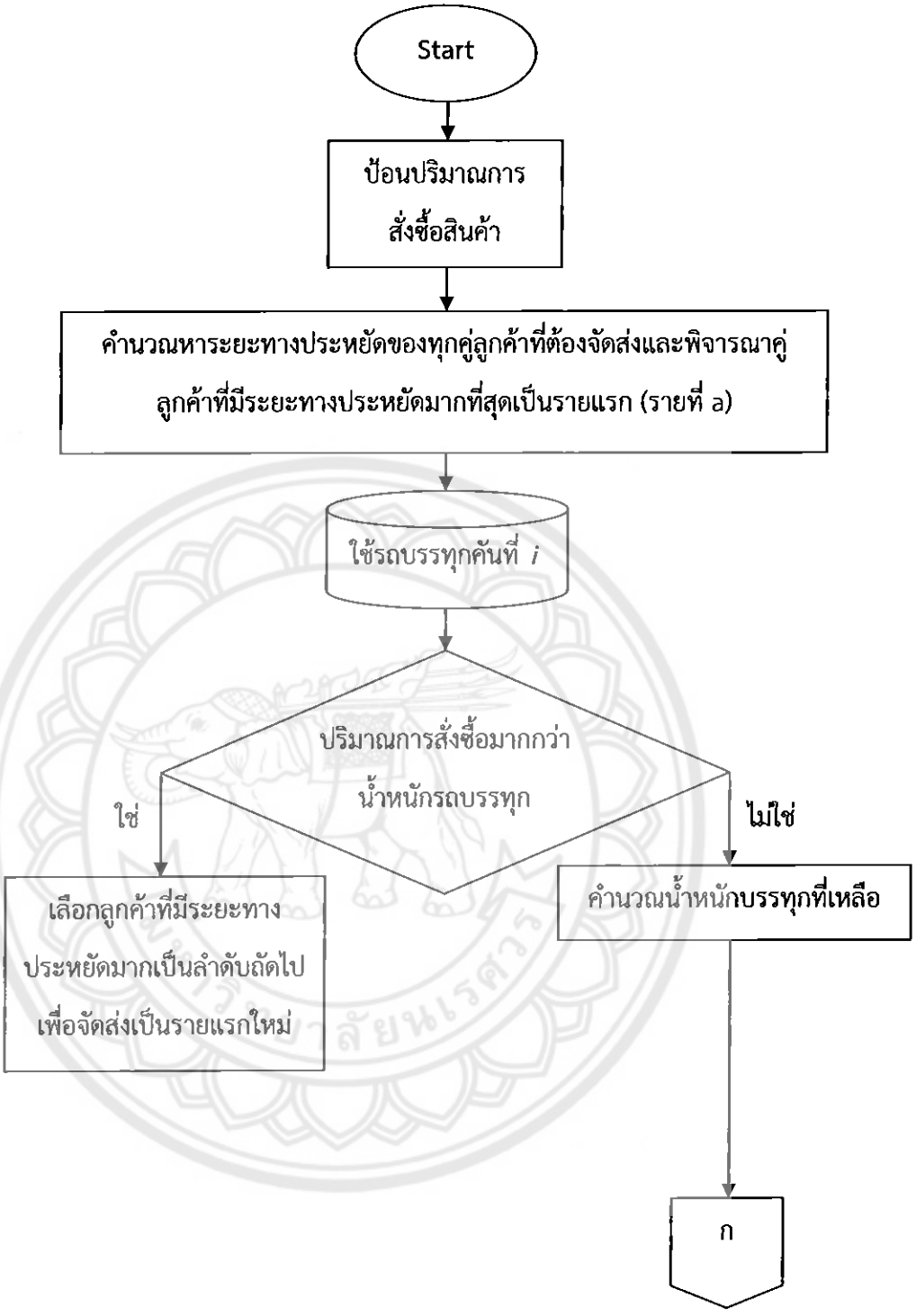
D_{i0} = ระยะทางจากจุดส่งสินค้า i ไปยังคลังสินค้า

D_{0j} = ระยะทางจากคลังสินค้าไปยังจุดส่งสินค้า j

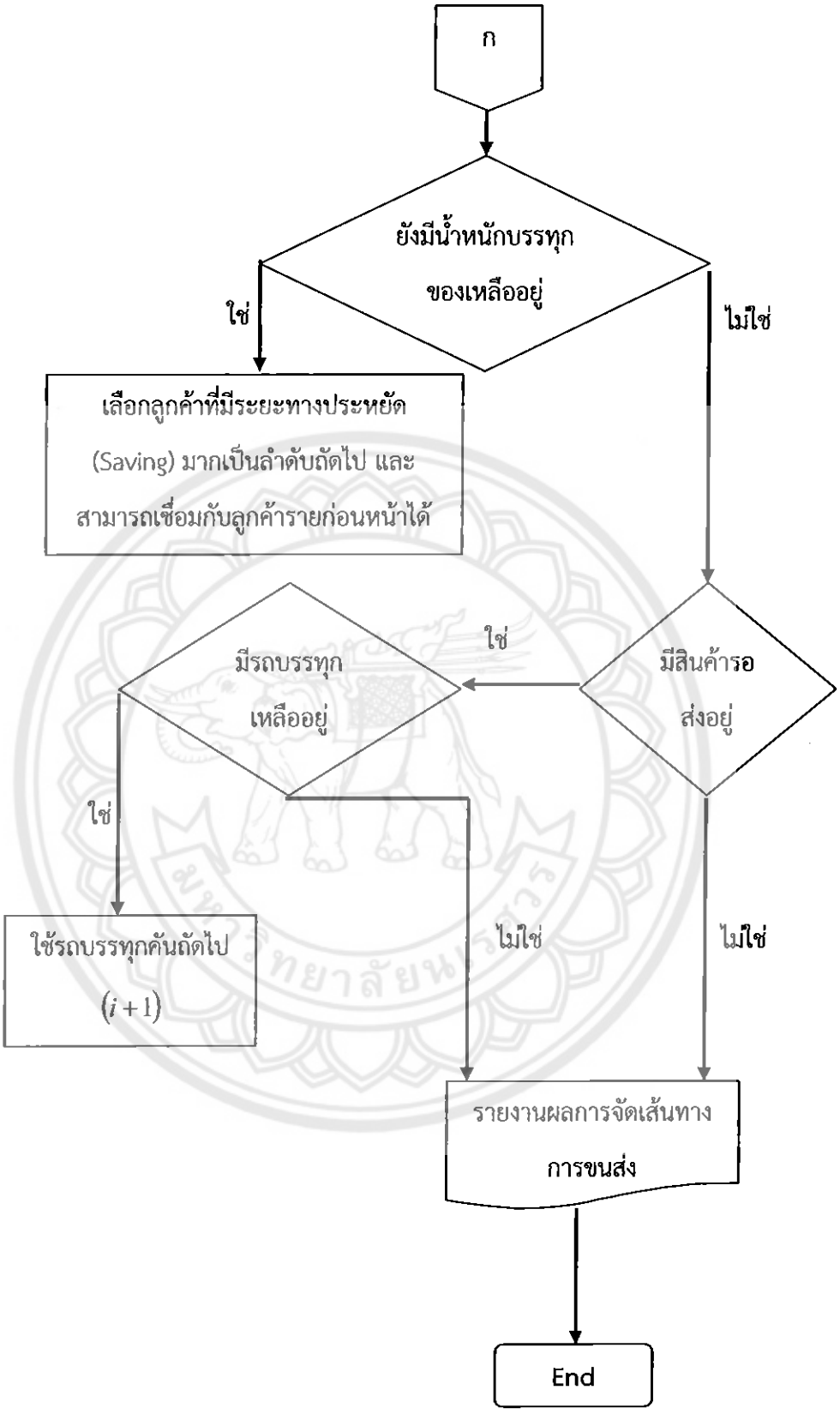
D_{ij} = ระยะทางจากจุดส่งสินค้า i ไปยังจุดส่งสินค้า j

ขั้นตอนที่ 2 ทำการเลือกคู่โหนดของลูกค้าที่มีค่าความประหยัดสูงสุด และยังไม่ถูกจัดเข้าไปอยู่ในเส้นทาง

ขั้นตอนที่ 3 เพิ่มลูกค้าลงในเส้นทาง โดยตรวจสอบเงื่อนไขของปัญหาว่าการรวมลูกค้าเข้าด้วยกันแล้ว จะทำให้เกิดความสามารถของยานพาหนะที่บรรทุกหรือไม่ หากพบว่าจำนวนสินค้าที่จุดใดๆ บนเส้นทางมีค่าเกินกว่าที่กำหนดไว้ ให้ทำการตัดลูกค้ารายล่าสุดออกจากเส้นทางและเชื่อมเส้นทางกลับไปยังคลังสินค้า และกลับไปทำขั้นตอนที่ 2 จนกระทั่งลูกค้าทุกรายถูกจัดให้อยู่ในเส้นทางการขนส่ง (Clarke and Wright, 1964) เพื่อให้สามารถเข้าใจถึงวิธีการหาเส้นทางการขนส่งสินค้าด้วยวิธีการเปรียบเทียบการประหยัดมากยิ่งขึ้น จึงมีการแสดงลำดับขั้นตอน ดังรูปที่ 2.5



รูปที่ 2.5 ขั้นตอนการหาเส้นทางการขนส่งสินค้าด้วยวิธีการเปรียบเทียบการประหยัด



รูปที่ 2.5 (ต่อ) ขั้นตอนการหาเส้นทางการขนส่งสินค้าด้วยวิธีการเปรียบเทียบการประหยัด

2.6.2 ตัวอย่างการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีเงื่อนไขเนื่องจากความจุ ด้วยวิธีการเปรียบเทียบการประหยัด

การจัดเส้นทางการเดินรถโดยรถมีความจุ 820 กิโลกรัม ด้วยวิธีการเปรียบเทียบการประหยัด (สมชาย ปฐมศิริ, 2551) ซึ่งสามารถแสดงในรูปแบบของตารางได้ดังตารางที่ 2.1 คำสั่งซื้อ และ ตารางที่ 2.2 แสดงระยะทางในการขนส่ง

ตารางที่ 2.1 คำสั่งซื้อ

ลูกค้า	ปริมาณการสั่งสินค้า (กิโลกรัม)
1	484
2	541
3	326
4	293
5	24
6	815
7	296
รวม	2781

ที่มา : สมชาย ปฐมศิริ, 2551

ตารางที่ 2.2 แสดงระยะทางในการขนส่ง

	บริษัท	ลูกค้า 1	ลูกค้า 2	ลูกค้า 3	ลูกค้า 4	ลูกค้า 5	ลูกค้า 6
ลูกค้า 1	19	-					
ลูกค้า 2	57	51	-				
ลูกค้า 3	51	10	49	-			
ลูกค้า 4	49	53	18	50	-		
ลูกค้า 5	4	25	30	11	68	-	
ลูกค้า 6	12	80	6	91	62	48	-
ลูกค้า 7	92	53	47	38	9	94	9

จากตารางที่ 2.2 จะได้เส้นทางที่ 1

บริษัท – ลูกค้ารายที่ 6 – บริษัท

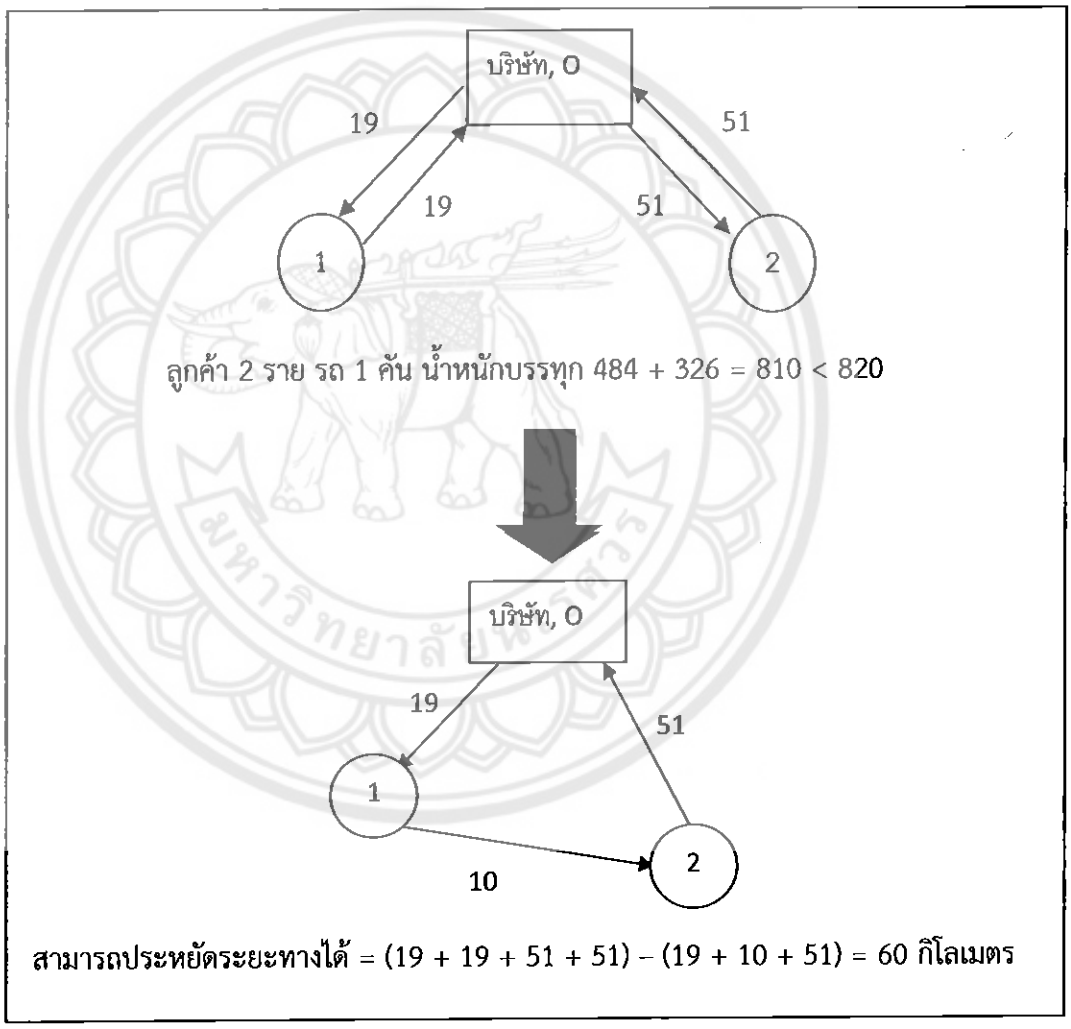
ปริมาณสินค้า 815 กิโลกรัม < 820 กิโลกรัม

ระยะทางรวม 12 + 12 = 24 กิโลเมตร

ยังเหลือลูกค้ารายที่ 1, 2, 3, 4, 5, 7

คำนวณเปรียบเทียบการประหยัดระยะทางแต่ละคู่ลูกค้า

จากรูปที่ 2.6 แสดงเส้นทางการเดินรถที่ปรับปรุงเส้นทางด้วยวิธีการเปรียบเทียบการประหยัดเพื่อให้มีระยะทางในการเดินทางที่สั้นที่สุด



รูปที่ 2.6 การปรับปรุงเส้นทางการเดินรถด้วยวิธีการเปรียบเทียบการประหยัด

การจัดเส้นทางการเดินรถที่มีการตรวจสอบน้ำหนักบรรทุกโดยรวมของสินค้าเทียบกับความจุของรถบรรทุก ดังแสดงตารางที่ 2.3 ซึ่งแสดงการจัดเส้นทางการเดินรถ

ตารางที่ 2.3 การจัดเส้นทางการเดินรถ

คู่ลูกค้า	รวมปริมาณสินค้า (กิโลกรัม)	ตรวจสอบน้ำหนักบรรทุกรวมโดยเทียบกับความจุ 820 กิโลกรัม	ระยะทางที่ประหยัด (กิโลเมตร)	จัดอันดับการประหยัดระยะทาง
1, 2	484 + 541 = 1,025	บรรทุกไม่ได้	ไม่ต้องพิจารณา	-
1, 3	484 + 326 = 810	บรรทุกได้	$C_{01} + C_{03} - C_{13} =$ 19 + 51 - 10 = 60	3
1, 4	484 + 293 = 777	บรรทุกได้	$C_{01} + C_{04} - C_{14} =$ 19 + 49 - 53 = 15	8
1, 5	484 + 24 = 508	บรรทุกได้	$C_{01} + C_{05} - C_{15} =$ 19 + 4 - 25 = -2	10
1, 7	484 + 296 = 780	บรรทุกได้	$C_{01} + C_{07} - C_{17} =$ 19 + 92 - 53 = 58	4
2, 3	541 + 326 = 867	บรรทุกไม่ได้	ไม่ต้องพิจารณา	-
2, 4	541 + 293 = 834	บรรทุกไม่ได้	ไม่ต้องพิจารณา	-
2, 5	541 + 24 = 565	บรรทุกได้	$C_{02} + C_{05} - C_{25} =$ 57 + 4 - 30 = 31	7
2, 7	541 + 296 = 837	บรรทุกไม่ได้	ไม่ต้องพิจารณา	-
3, 4	326 + 293 = 619	บรรทุกได้	$C_{03} + C_{04} - C_{34} =$ 51 + 49 - 50 = 50	5
3, 5	326 + 24 = 350	บรรทุกได้	$C_{03} + C_{05} - C_{35} =$ 51 + 4 - 11 = 44	6
3, 7	326 + 296 = 622	บรรทุกได้	$C_{03} + C_{07} - C_{37} =$ 51 + 92 - 38 = 105	2

ตารางที่ 2.3 (ต่อ) การจัดเส้นทางการเดินรถ

คู่ลูกค้า	รวมปริมาณสินค้า (กิโลกรัม)	ตรวจสอบน้ำหนักบรรทุกรวมโดยเทียบกับความจุ 820 กิโลกรัม	ระยะทางที่ประหยัด (กิโลเมตร)	จัดอันดับการประหยัดระยะทาง
4, 5	293 + 24 = 317	บรรทุกได้	$C_{04} + C_{05} - C_{45} =$ 49 + 4 - 68 = -15	11
4, 7	293 + 296 = 589	บรรทุกได้	$C_{04} + C_{07} - C_{47} =$ 49+92-9 =132	1
5, 7	24 + 296 = 320	บรรทุกได้	$C_{05} + C_{07} - C_{57} =$ 4 + 92 - 94 = 2	9

จากตารางที่ 2.4 แสดงเส้นทางการเดินรถที่มีระยะทางรวมต่ำที่สุด และความจุรวมของสินค้าที่ต้องส่งให้กับลูกค้าไม่เกินความจุของรถบรรทุกที่กำหนด 820 กิโลกรัม

ตารางที่ 2.4 ผลการจัดเส้นทาง ตรวจสอบคำตอบ 4 เส้นทาง

เส้นทางเดินรถ	น้ำหนักบรรทุก (กิโลกรัม)	ระยะทางทั้งหมด (กิโลเมตร)
1. บริษัท – ลูกค้า 6 - บริษัท	815	12 + 12 = 24
2. บริษัท – ลูกค้า 1 – ลูกค้า 3 - บริษัท	484 + 326 = 810	19 + 10 + 51 = 80
3. บริษัท – ลูกค้า 1 – ลูกค้า 3 - บริษัท	484 + 326 = 810	19 + 10 + 51 = 80
4. บริษัท – ลูกค้า 4 – ลูกค้า 7 - บริษัท	293 + 296 = 589	49 + 9 + 92 = 150

2.7 วิธีเมตาฮิวริสติก (Metaheuristic Method)

เมตาฮิสริสติก หมายถึง ชุดของลำดับขั้นการแก้ปัญหาแบบฮิวริสติกชนิดหนึ่งที่สามารถนำหลักการเดียวกันไปใช้แก้ปัญหาได้หลากหลายปัญหาซึ่งในปัจจุบัน วิธีการออกแบบฮิวริสติกโดยอาศัยหลักการทางเมตาฮิวริสติกนี้ ได้รับความนิยมเป็นอย่างสูง เนื่องจากคำตอบที่ได้จากวิธีการนี้ให้ผลที่ดี แก้ปัญหาได้รวดเร็ว และใช้งานง่าย โดย Blum and Roli (2003) ได้กล่าวถึงหลักการเบื้องต้นของเมตาฮิวริสติกไว้ ดังต่อไปนี้

2.7.1 เมตาฮิวริสติกมีระเบียบวิธีในการค้นหาคำตอบที่ดีภายในพื้นที่ของคำตอบที่เป็นไปได้ (Feasible Region)

2.7.2 เมตาฮิวริสติกมีจุดประสงค์เพื่อหาคำตอบที่ดีที่สุด หรือคำตอบที่ใกล้เคียงคำตอบที่ดีที่สุดภายในระยะเวลาอันสั้น

2.7.3 วิธีการทางเมตาฮิวริสติกจะมีทั้งแบบง่ายไม่ซับซ้อน เช่น การปรับปรุงคำตอบเฉพาะที่ หรือแบบที่ยุ่งยากซับซ้อนมากกว่า เช่น วิธีระบบมด (Ant System) วิธีการเชิงพันธุกรรม (Genetic Algorithm) วิธีการค้นหาต้องห้าม (Tabu Search) วิธีการเลียนแบบการอบอ่อน (Simulated Annealing) เป็นต้น

2.7.4 เมตาฮิวริสติกเป็นขั้นตอนการประมาณคำตอบ

2.7.5 เมตาฮิวริสติกอาจจะเกิดจากการรวมหลากหลายเทคนิค เพื่อค้นหาคำตอบที่ดีที่สุดภายในพื้นที่คำตอบที่เป็นไปได้

2.7.6 เมตาฮิวริสติกมีระเบียบขั้นตอนมาตรฐานที่แน่นอน แม้ว่าเมื่อนำไปประยุกต์ใช้ในปัญหาที่แตกต่างกันจะมีรายละเอียดของขั้นตอนย่อยที่แตกต่างกัน แต่ฮิวริสติกสำหรับปัญหาแต่ละปัญหาต้องดำเนินการตามขั้นตอนหลักของเมตาฮิวริสติกดั้งเดิม

2.7.7 เมตาฮิวริสติกต้องสามารถใช้ได้กับปัญหาที่หลากหลาย

2.7.8 เมตาฮิวริสติกอาจจะมีลักษณะเป็นคำบรรยายโดยย่อก็ได้ หรือไม่จำเป็นต้องมีหลักการทางคณิตศาสตร์

2.7.9 ปัจจุบันนี้ เมตาฮิวริสติกใช้ความจำชั่วคราวมากขึ้นในการจำคำตอบเดิม เพื่อค้นหาคำตอบที่ไม่ซ้ำเดิมหรือแตกต่างไปจากเดิม เช่น วิธีการค้นหาต้องห้าม วิธีระบบมด เป็นต้น

ซึ่งเนื่องจากวิธีเมตาฮิวริสติกเป็นวิธีที่ได้จากการพัฒนา และดัดแปลงมาจากวิธีฮิวริสติกเพื่อให้มีความยืดหยุ่นในการหาผลเฉลยของปัญหาการตัดสินใจใดๆ ที่มีความซับซ้อนและมีตัวแปรการตัดสินใจจำนวนมากได้อย่างรวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้น ถึงแม้ผลเฉลยที่ได้อาจไม่ใช่ผลเฉลยที่ให้ค่าที่ดีที่สุด หรือไม่สามารถรับประกันผลเฉลยที่ดีในทุกครั้งที่ทำการประมวลผลได้ แต่ผลเฉลยที่ได้เป็นที่ยอมรับและค้นหาได้ภายในระยะเวลาอันเหมาะสม จึงเป็นที่นิยมอย่างกว้างขวาง ซึ่งในวิจัยนี้ได้นำวิธีเมตาฮิวริสติกวิธีหนึ่งมาใช้ในการแก้ปัญหาสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ คือ วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

2.8 ตัวอย่างการใช้วิธีการเมตาฮิวริสติกที่ใช้ในการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะ

2.8.1 ขั้นตอนวิธีการค้นหาแบบทาบู (Tabu Search)

วิธีการค้นหาแบบทาบู หมายถึง การห้ามหรือการป้องกันขั้นตอนวิธีการเข้าไปยังผลเฉลยข้างเคียงที่ไม่ถูกต้อง โดยแนวคิดสำคัญ คือ การเพิ่มความฉลาด ซึ่งใช้ความทรงจำของคอมพิวเตอร์มาเกี่ยวข้อง คอมพิวเตอร์จะเรียนรู้จากรอบการวนซ้ำที่ผ่านมา (Search History) ในการแนะนำ หรือ

บอกทิศทางของผลเฉลยที่ดีหรือดีที่สุดในรอบการกระทำซ้ำถัดไปข้างหน้า ซึ่งวิธีการค้นหาแบบทาบูเป็นวิธีที่ค่อนข้างได้รับความนิยมอย่างมาก เพราะมีโครงสร้างของขั้นตอนวิธีการที่ไม่ซับซ้อนมากนัก และจะแก้ปัญหาการยอมรับค่าเหมาะสุดเฉพาะที่ โดยป้องกันไม่ให้มีการเกิดขึ้น (Pro – Active) ของผลเฉลยข้างเคียง

2.8.2 ขั้นตอนวิธีการจำลองการอบอ่อน (Simulated Annealing Algorithm)

ในการหาผลเฉลยของปัญหาการตัดสินใจ เริ่มต้นครั้งแรกเมื่อประมาณ ค.ศ. 1980 ลักษณะเด่นของวิธีการนี้ คือ มีขั้นตอนวิธีการหาผลเฉลยที่ง่าย และมีประสิทธิภาพ ซึ่งสามารถหาผลเฉลยที่ดีได้ภายในเวลาอันรวดเร็ว แนวคิดพื้นฐานของวิธีการจำลองการอบอ่อนได้ถูกตีพิมพ์ครั้งแรกในวารสารวิชาการโดย Metropolis et al.. (1953) ซึ่งเป็นการใช้ขั้นตอนวิธีการจำลองการควบคุมการเย็นตัวลงของวัสดุในอ่างความร้อน (Heat Bath) ซึ่งเรียกขั้นตอนนี้ว่าการอบอ่อน (Annealing) วัสดุจะถูกให้ความร้อนจนกระทั่งถึงจุดหลอมเหลว ต่อจากนั้นเมื่อเวลาผ่านไปอุณหภูมิของวัสดุจะค่อยๆ ลดลง และทำให้วัสดุมีความแข็งเมื่อเย็นตัวลง โครงสร้างคุณสมบัติของวัสดุที่เย็นตัวลงนี้จะขึ้นอยู่กับอัตราการทำให้วัสดุนั้นเย็นตัวลง (Rate of Cooling)

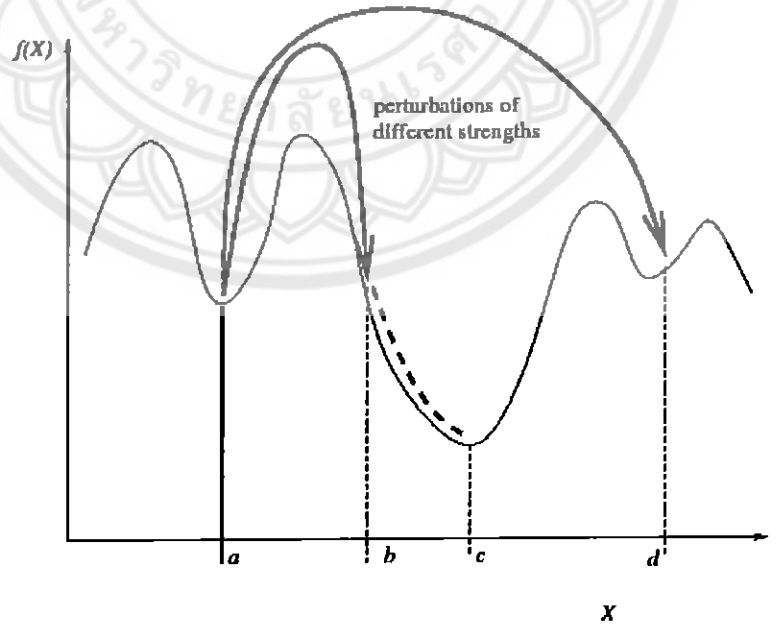
ขั้นตอนวิธีการของ Metropolis นี้ ได้จำลองการเปลี่ยนแปลงพลังงานของระบบที่แปรผันขั้นตอนโดยกระบวนการเย็นตัวจนกระทั่งการเปลี่ยนแปลงนั้นเข้าสู่สภาวะคงที่ ประมาณ 30 ปีต่อมาได้มีการนำขั้นตอนวิธีการดังกล่าวมาใช้ในการแก้ปัญหาการตัดสินใจที่ต้องการหาค่าเหมาะสมที่ต่ำสุดหรือสูงสุด ซึ่งเปรียบเสมือนว่า ผลเฉลยที่ได้จากวิธีการจำลองการอบอ่อนจะค่อยๆ เข้าสู่ผลเฉลยที่ดีที่สุดเหมือนในช่วงเวลาการเย็นตัวลงของวัสดุ

2.9 การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ เป็นเมตาฮิวริสติกที่พัฒนามาจากการค้นหาคำตอบเฉพาะที่ (Basic Local Search : BLS) โดยที่แนวคิดจากการค้นหาคำตอบเฉพาะที่ คือ ค้นหาจุดที่ดีที่สุดในพื้นที่หนึ่งจำกัดในพื้นที่ที่เป็นไปได้ของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ นำข้อดีของการค้นหาคำตอบเฉพาะที่ มาใช้คือหาคำตอบที่ดีที่สุดในแต่ละพื้นที่ หลังจากนั้นจะรบกวนคำตอบ (Perturbation) เพื่อให้ออกจากพื้นที่เดิม (Escape) แล้วค้นหาคำตอบที่ดีที่สุดในพื้นที่ใหม่ ถ้าจินตนาการเหมือนการหาอาหารของกบ สมมติให้กบกระโดดไปหากินแมลงที่บริเวณหนึ่ง หลังจากกินแมลงบริเวณนี้ทั้งหมดแล้ว (หมายถึงบริเวณรอบตัวกบ) กบจะกระโดดไปหาพื้นที่ใกล้ๆ นั้น และกินแมลงรอบๆ บริเวณใหม่นั้นจนหมด และกระโดดไปยังพื้นที่ใกล้ๆ ต่อเรื่อยๆ การที่กบกระโดดไปและหาอาหารกินบริเวณคำตอบ การกระโดดไปหาพื้นที่ใหม่ใกล้ๆ พื้นที่เดิมคือการรบกวนคำตอบ เช่นเดียวกับหลักการของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ ที่มีการหาคำตอบบริเวณรอบๆซึ่งก็คือการค้นหาคำตอบเฉพาะที่ และรบกวนคำตอบเดิมเพื่อให้ได้พื้นที่ใหม่ในการค้นหาคำตอบเฉพาะที่จากนั้นทำซ้ำไปเรื่อยๆ จนกระทั่งผู้ออกแบบฮิวริสติกพอใจ จึงหยุดการรบกวนและการค้นหาคำตอบ

เฉพาะที่ จากหลักการพื้นฐานดังกล่าว คุณภาพของเมตาฮิวริสติกที่ได้จากการประยุกต์ใช้การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ จะขึ้นอยู่กับ 2 ประการหลัก คือ คุณภาพของการค้นหาหรือปรับปรุงคำตอบเฉพาะที่ (Local Search : LS) และเทคนิคการรบกวนคำตอบ (Perturbation Techniques)

ประการแรก การใช้การค้นหาคำตอบที่ที่ดีจะทำให้ค่าที่ดีที่สุดในแต่ละพื้นที่ได้รวดเร็วและได้ผลดี ถ้าใช้วิธีการค้นหาคำตอบเฉพาะที่ที่ไม่เหมาะสมจะไม่สามารถหาค่าที่ดีที่สุดในแต่ละพื้นที่ได้ ทำให้ไม่ได้คำตอบที่เป็นคำตอบที่ดีที่สุดในพื้นที่ย่อย (Local Optimum) ประการที่ 2 คือ การรบกวนที่ไม่มากไม่น้อยจนเกินไป ถ้ามากเกินไปจะทำให้ข้ามบางพื้นที่ที่มีคำตอบที่ดีไป เหมือนการหาอาหารของกบ ถ้ากระโดดจากพื้นที่เดิมไกลเกินไป อาจจะลืมหาอาหารบริเวณที่มีแมลงชุกชุมที่ไม่ห่างจากพื้นที่ที่ค้นหาในปัจจุบัน ทำให้การค้นหาแบบนี้เรียกว่า การค้นหาแบบเดินสุ่ม (Random Walk) ซึ่งจะเหมือนกับการเริ่มต้นหาคำตอบแรกเสมอ ไม่มีการเรียนรู้ของฮิวริสติก ซึ่งจะทำให้ได้คำตอบจากวิธีที่พัฒนาที่ไม่ดีตามที่คาดหวังไว้ แต่ถ้าการรบกวนน้อยจนเกินไปจะทำให้ไม่สามารถออกจากพื้นที่เก่าได้ เหมือนกับกบที่ขาหัก กระโดดไปหาอาหารบริเวณอื่นไม่ได้ ต้องการหาอาหารบริเวณเดิมๆ ไปเรื่อยๆ จนอาหารหมด ซึ่งก็จะได้คำตอบสุดท้ายที่เป็นเพียงคำตอบที่ดีที่สุดในพื้นที่ย่อยเท่านั้น ข้อดีของวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ คือ มีความง่าย เร็ว และตัวแปรต่างๆ น้อยไม่ซับซ้อนในการคำนวณ (ระพีพันธ์ ปิตาคะโส, 2554) และเพื่อให้เข้าใจหลักการในการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำมากขึ้น จึงสามารถอธิบายได้จากรูปที่ 2.7 ซึ่งแสดงการค้นหาพื้นที่คำตอบที่ดีที่สุดของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ



รูปที่ 2.7 การค้นหาพื้นที่คำตอบที่ดีที่สุดของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

ที่มา : http://lion.disi.unitn.it/reactive-search/thebook/node10.html

จากรูปที่ 2.7 การค้นหาพื้นที่คำตอบที่ดีที่สุดของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ จะเห็นได้ว่าจุด a จะเป็นพื้นที่คำตอบที่ดีที่สุดในการค้นหาพื้นที่คำตอบนั้น จากนั้นก็ได้มีการรบกวนคำตอบ (Perturbation) เพื่อค้นหาคำตอบที่ดีที่สุดในพื้นที่ใหม่ ซึ่งในการรบกวนคำตอบ ถ้ารบกวนคำตอบมากเกินไป (จุด d) ก็จะทำให้ข้ามบางพื้นที่ที่มีคำตอบที่ดีไป (จุด c) และถ้ามีการรบกวนคำตอบน้อยเกินไปก็จะไม่สามารถออกจากพื้นที่เก่าได้ แต่ถ้ามีการรบกวนคำตอบที่เหมาะสม (จุด b) ก็จะสามารถค้นหาคำตอบที่ดีที่สุด (จุด c) ในพื้นที่ใหม่ได้

รหัสเทียมของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำในการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำนี้ สามารถเขียนรหัสเทียมขึ้นมาได้ ซึ่งรหัสเทียมจะเป็นส่วนผสมของการใช้ภาษาธรรมชาติและโปรแกรมภาษา เพื่อใช้ในการสื่อสารระหว่างผู้ใช้หลายกลุ่มที่มีความต้องการใช้ขั้นตอนวิธีต่างกันออกไป รหัสเทียมไม่มีกฎในการเขียนตายตัว มีลักษณะคล้ายกับภาษาคอมพิวเตอร์ โดยมากขึ้นอยู่กับความถนัดของผู้ใช้ แต่มีข้อตกลงบางอย่างร่วมกันเป็นสากล ส่วนประกอบที่สำคัญของรหัสเทียมได้แก่ ชื่อ คำสั่งกำหนดงาน คำสั่งควบคุม กลุ่มของคำสั่ง และข้อบันทึกหรือคำอธิบาย ซึ่งเกณฑ์ในการเขียนรหัสเทียม มีดังนี้

2.9.1 ประโยคคำสั่ง เขียนเป็นภาษาอังกฤษอย่างง่าย

2.9.2 ประโยคคำสั่งหนึ่งๆ จะเขียนต่อหนึ่งบรรทัดเท่านั้น

2.9.3 ควรใช้ย่อหน้าให้เป็นประโยชน์เพื่อแยกคำเฉพาะ (Keywords) ได้อย่างชัดเจน

2.9.4 แต่ละประโยคคำสั่งให้เขียนลำดับจากบนลงล่างโดยมีเพียงทางเข้าทางเดียวและมีทางออกทางเดียวเท่านั้น

2.9.5 กลุ่มของประโยคคำสั่งต่างๆ อาจจัดรวมกลุ่มเข้าด้วยกันในรูปแบบของโมดูล แต่ต้องกำหนดชื่อโมดูลเหล่านั้นด้วย เพื่อให้สามารถเรียกใช้โมดูลเหล่านั้นได้

และเพื่อให้เข้าใจรหัสเทียมของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำมากยิ่งขึ้น จึงได้แสดงตัวอย่างรหัสเทียม ดังรูปที่ 2.8 ซึ่งแสดงรหัสเทียมของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

```
Procedure ILS ( )
    initial := InitialSolution
    s := LocalSearch (initial)
    best := s
    while not Terminate ( ) do
        s' := Perturbate(s)
        s'' := LocalSearch(s')
        if f(s'') < f(best) then
            best := s''
            s := best
        endif
    endwhile
    return best
end
```

รูปที่ 2.8 รหัสเทียมของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

จากรูปที่ 2.8 จะแสดงถึงรหัสเทียมของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ ซึ่งรหัสเทียมนี้จะเป็นการหาค่าต่ำสุด (Minimization Problem) ของแต่ละพื้นที่ โดยจะเริ่มจากสร้างคำตอบเริ่มต้นจากการสุ่มคำตอบจากพื้นที่ที่เป็นไปได้มา 1 ค่า จากนั้นทำการปรับปรุงคำตอบในพื้นที่นั้น จนได้คำตอบที่ดีที่สุดในพื้นที่เป็น s และให้ $s = best$ เมื่อได้คำตอบที่ดีที่สุดในพื้นที่นั้นแล้วก็ทำการรบกวนคำตอบ s ให้ออกจากพื้นที่เดิม ซึ่งจะได้เป็น s' แล้วทำการปรับปรุงคำตอบเพื่อให้ได้คำตอบที่ดีที่สุดในพื้นที่นั้น (s'') จากนั้นนำค่าคำตอบที่ดีที่สุดในแต่ละพื้นที่ $best$ และ s'' มาเปรียบเทียบคำตอบกัน ถ้าคำตอบ s'' ให้คำตอบที่ดีกว่า $best$ นั่นคือ $f(s'') < f(best)$ ให้ $best = s''$ และให้ $s = best$ ซึ่งจะสิ้นสุดเงื่อนไขในรอบแรก จากนั้นก็ทำซ้ำไปเรื่อยๆ จนกระทั่งถึงเงื่อนไขการหยุดที่ระบุไว้

2.10 หลักการและทฤษฎีของภาษา Visual Basic for Application

Visual Basic for Application (VBA) เป็นภาษาที่ใช้สำหรับเขียนโปรแกรมควบคุมการทำงานของโปรแกรม Microsoft Office เช่น Microsoft Word หรือ Microsoft Excel เป็นต้น ซึ่งในที่นี้จะกล่าวเฉพาะการเขียนโปรแกรมด้วยภาษา Visual Basic for Application ใน Microsoft Excel เท่านั้น สำหรับองค์ประกอบพื้นฐานของการเขียนโปรแกรมภาษา Visual Basic for Application ใน Microsoft Excel มีส่วนสำคัญที่ควรรู้ เพื่อให้สามารถเขียนคำสั่งควบคุมส่วนต่างๆ ได้อย่างถูกต้อง ดังนี้

2.10.1 Workbook

เป็นส่วนที่ใหญ่ที่สุด ไว้สำหรับเก็บข้อมูลของไฟล์ทั้งหมด ภายในจะประกอบไปด้วย Worksheet ต่างๆ

2.10.2 Worksheet

เป็นแผ่นงานที่มีลักษณะเหมือนตารางที่มีแถวและคอลัมน์ตัดกัน ซึ่งส่วนที่ตัดกันเป็นช่องๆ ไว้สำหรับป้อนข้อมูลเรียกว่า เซลล์

2.10.3 Modules

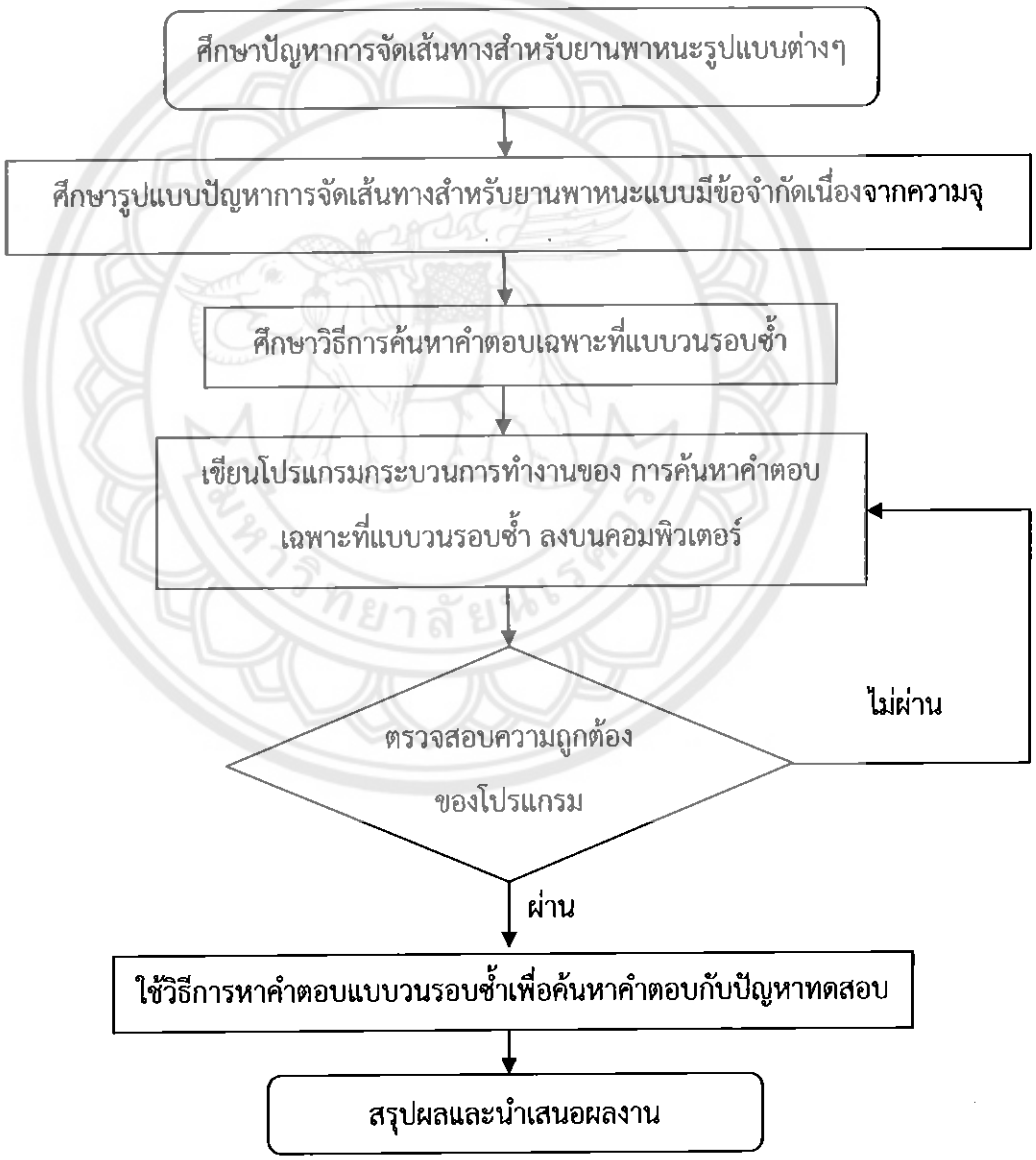
เป็นส่วนที่ไว้สำหรับเก็บคำสั่งรหัสรวมทั้ง Macro ต่างๆ ที่บันทึกไว้ เพื่อให้สามารถเรียกขึ้นมาใช้งานได้ในภายลัง

ข้อดีของการเขียนโปรแกรมด้วยภาษา Visual Basic for Application ใน Microsoft Excel เช่น การทำให้ Excel ทำงานซ้ำๆ มากกว่าหนึ่งครั้ง การสั่งให้ Excel ทำงานตามเงื่อนไขที่เกิดขึ้น การเชื่อมต่อฐานข้อมูลโดยมีเงื่อนไขในการค้นหาข้อมูล การสร้างเมนูเพิ่มขึ้นใน Excel และการสร้างระบบงานใน Excel ที่เหมือนโปรแกรมสำเร็จรูป ดังนั้น Visual Basic for Application จึงมีการประยุกต์ใช้กันอย่างแพร่หลาย เพราะเป็นโปรแกรมพื้นฐานที่มีใช้ทั่วไปในเครื่องคอมพิวเตอร์แทบจะทุกเครื่อง โดยเฉพาะในงานวางแผน และบริหาร จะช่วยให้ทำงานได้ง่ายและรวดเร็วยิ่งขึ้น

ในโครงงานวิจัยนี้ได้นำเอา Visual Basic for Application มาใช้ในการสร้างฟังก์ชันบน Microsoft Excel โดยนำมาสร้างฟังก์ชันคำนวณหาระยะทางในการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุเพื่อให้ได้ระยะทางรวมในการขนส่งสินค้าที่น้อยที่สุด โดยจะสร้างส่วนติดต่อกับผู้ใช้ (User From) เพื่อให้ผู้ใช้กรอกข้อมูลที่ต้องการในการคำนวณ ซึ่งมีหลักการในการเชื่อมโยงข้อมูลในแผ่นงาน Excel มาใช้ในการคำนวณร่วมกับข้อมูลส่วนติดต่อกับผู้ใช้ และส่งให้แสดงผลการคำนวณหาค่าระยะทางรวมในการขนส่งสินค้าที่น้อยที่สุดบน Microsoft Excel

บทที่ 3
วิธีการดำเนินโครงงาน

หลังจากที่ได้มีการศึกษาค้นคว้าข้อมูล และงานวิจัยที่เกี่ยวกับปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดด้านความจุ ดังที่กล่าวไว้ในบทที่ 1 และบทที่ 2 แล้ว และเพื่อทำให้การศึกษาวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำสามารถเข้าใจได้ง่ายยิ่งขึ้น จึงมีการดำเนินงานที่เป็นลำดับขั้นตอนดังรูปที่ 3.1 ซึ่งแสดงแสดงขั้นตอนการดำเนินงานวิจัยของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดด้านความจุ



รูปที่ 3.1 แสดงขั้นตอนการดำเนินงานวิจัยของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดด้านความจุ

3.1 ศึกษาปัญหาการจัดเส้นทางสำหรับยานพาหนะรูปแบบต่างๆ

ศึกษาปัญหาการจัดเส้นทางสำหรับยานพาหนะรูปแบบต่างๆ โดยศึกษาจากงานวิจัยของ Toth and Vigo (2001)

3.2 ศึกษารูปแบบปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ

3.2.1 ศึกษารูปแบบปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ โดยศึกษาจากงานวิจัยของ Toth and Vigo (2001)

3.2.2 ศึกษาแบบจำลองทางคณิตศาสตร์ของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ โดยมีข้อตกลงเบื้องต้น ดังนี้

3.2.2.1 ลูกค้าแต่ละรายจะได้รับสินค้าจากยานพาหนะเพียงคันเดียวเท่านั้น

3.2.2.2 จำนวนยานพาหนะที่วิ่งออกจากคลังสินค้าเพื่อส่งสินค้าให้กับลูกค้าจะต้องเท่ากับจำนวนยานพาหนะที่วิ่งกลับเข้ามายังคลังสินค้า

3.2.2.3 ยานพาหนะแต่ละคันจะใช้เส้นทางใดเส้นทางหนึ่งเท่านั้น ในการขนส่งสินค้าให้ลูกค้า

3.2.2.4 ยานพาหนะสามารถขนส่งสินค้าให้กับลูกค้าได้ไม่เกินความจุที่ยานพาหนะสามารถบรรทุกได้

3.3 ศึกษาวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

3.3.1 ศึกษาฟังก์ชั่นการทำงาน การกำหนดตัวแปรต่างๆ

3.3.2 ศึกษาวิธีการทำงานของวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

3.4 เขียนโปรแกรมกระบวนการทำงานของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำลงบนคอมพิวเตอร์

ทำการเขียนโปรแกรมกระบวนการทำงานของ การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ ลงบนคอมพิวเตอร์ โดยใช้ภาษา Visual Basic for Applications บน Microsoft Excel

3.5 ตรวจสอบความถูกต้องของโปรแกรม

ทำการตรวจสอบความถูกต้องของแบบโปรแกรมหลังจากที่ได้เขียนโปรแกรมกระบวนการทำงานของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ เพื่อใช้ในการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุเรียบร้อยแล้ว

3.6 ใช้วิธีการหาคำตอบแบบวนรอบซ้ำเพื่อค้นหาคำตอบกับปัญหาทดสอบ

ประมวลผลคำตอบที่ได้จากวิธีการหาคำตอบแบบวนรอบซ้ำ โดยทดสอบกับโจทย์ปัญหาที่สร้างขึ้น

3.7 สรุปผลและนำเสนอผลงาน

นำผลลัพธ์ที่ได้จากโปรแกรมมาวิเคราะห์ พิจารณาความเป็นไปได้ของคำตอบว่าเหมาะสมหรือไม่ แล้วจึงสรุปผล จัดทำรายงาน และนำเสนอ

บทที่ 4
การใช้โปรแกรมและผลการทดสอบ

ในบทนี้จะกล่าวถึงผลการออกแบบการเขียนโปรแกรมคอมพิวเตอร์ด้วยโปรแกรม Visual Basic for Applications บน Microsoft Excel และผลการทดสอบโปรแกรมการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก รวมทั้งทดสอบประสิทธิภาพของโปรแกรมที่ทำการออกแบบ นอกจากนี้ยังเป็นการเปรียบเทียบและวิเคราะห์ผลที่ได้กับงานวิจัยอื่นที่กล่าวมาในบทที่ 3

ซึ่งจะสามารถแบ่งหัวข้อของผลการทดลองและการวิเคราะห์ได้ ดังต่อไปนี้

4.1 ออกแบบวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำสำหรับปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ

4.2 ความแตกต่างของ Local Search Type 1, Local Search Type 2 และ Iterated Local Search

4.3 การใช้งานของโปรแกรม การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

4.4 ผลการทดสอบโปรแกรมและวิเคราะห์ผลการทดลองในปัญหาขนาดเล็ก

4.5 ผลการทดสอบโปรแกรมและวิเคราะห์ผลการทดลองในปัญหาขนาดกลาง

4.6 ผลการทดสอบโปรแกรมและวิเคราะห์ผลการทดลองในปัญหาขนาดใหญ่

4.7 การวิเคราะห์ผลการทดลอง

4.1 ออกแบบวิธีการค้นหาคำตอบเฉพาะที่สำหรับปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ

ในขั้นตอนนี้ผู้วิจัยได้ทำการออกแบบวิธีแก้ไขปัญหาโดยวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ แล้วนำไปเขียนโปรแกรมคอมพิวเตอร์ด้วยภาษา Visual Basic for Applications บนโปรแกรม Microsoft Excel การดำเนินการออกแบบจำลองนี้ จะมีกระบวนการหลักอยู่ 4 กระบวนการ คือ การหาคำตอบเริ่มต้นที่ได้จากการสุ่ม การหาคำตอบใหม่ในย่านคำตอบเดิม กระบวนการจัดเรียงรอบรถ และออกแบบเมตาฮิวริสติกสามารถอธิบายได้โดยใช้ตัวอย่างการทำงานดังต่อไปนี้

4.1.1 การหาคำตอบเริ่มต้นที่ได้จากการสุ่ม

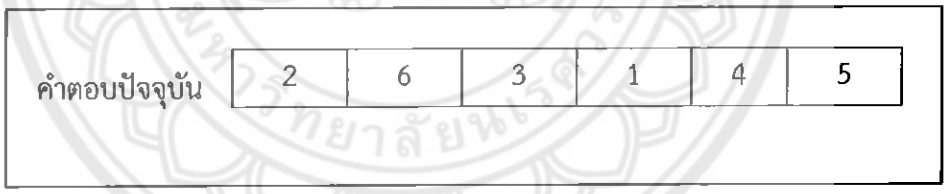
สำหรับการหาคำตอบเริ่มต้นในที่นี้ จะได้จากการนำเลข 1, 2, 3,..., n (เมื่อ n = จำนวนลูกค้าทั้งหมด) มาสุ่มเพื่อสลับตำแหน่ง จะทำให้ได้คำตอบเริ่มต้นเป็นเลขสุ่มที่ไม่ซ้ำกันตั้งแต่ 1 - n

4.1.2 การหาคำตอบใหม่ในย่านคำตอบเดิม

4.1.2.1 การค้นหาคำตอบแบบสลับ 1 (SWAP1)

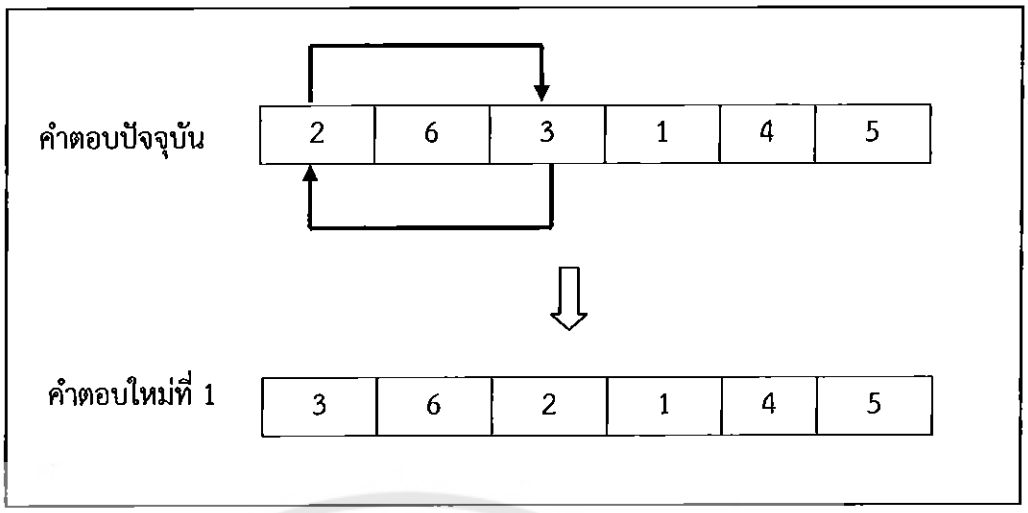
การค้นหาคำตอบแบบสลับ 1 เป็นวิธีการหาคำตอบที่จะนำเอาตัวแทนคำตอบปัจจุบันมาใช้ในการค้นหาคำตอบแบบสลับ 1 โดยจะเริ่มจากการสุ่มตำแหน่งในคำตอบปัจจุบันขึ้นมา 1 ตำแหน่ง เพื่อให้ตำแหน่งที่สุ่มมาได้นี้ทำการสลับที่กับตำแหน่งหลัก ซึ่งถ้าตำแหน่งที่สุ่มมาได้นั้นเป็นตำแหน่งเดียวกันกับตำแหน่งหลักก็จะทำการสุ่มตำแหน่งขึ้นมาจนกระทั่งตำแหน่งใหม่ที่สุ่มมาได้นั้นไม่ซ้ำกับตำแหน่งหลัก ซึ่งตำแหน่งหลักในที่นี้ก็ คือ ตำแหน่งที่จะใช้ยึดเป็นตำแหน่งหลักในการสลับที่กับตำแหน่งใหม่ที่สุ่มมาได้จากคำตอบปัจจุบัน โดยจะให้ตำแหน่งหลักนี้เริ่มที่ตำแหน่งที่ 1 หลักจากทำการสลับที่ระหว่างตำแหน่งหลักตำแหน่งที่ 1 กับตำแหน่งที่สุ่มมาได้แล้ว ก็จะได้เป็นคำตอบใหม่ที่ 1 จากนั้นก็ทำการหาคำตอบใหม่ที่ 2 โดยการสุ่มหาตำแหน่งจากคำตอบปัจจุบันเช่นเดียวกัน แต่ตำแหน่งที่สุ่มมานี้จะใช้ในการสลับที่กับตำแหน่งหลักตำแหน่งที่ 2 ซึ่งจะทำการสลับที่ในรูปแบบนี้ไปเรื่อยๆ โดยที่การหาคำตอบใหม่ที่ n สำหรับตำแหน่งที่สุ่มขึ้นมาได้จะต้องทำการสลับที่กับตำแหน่งหลักที่ n (เมื่อ n = จำนวนลูกค้าทั้งหมด) ดังนั้นคำตอบใหม่ที่ได้จะต้องมีจำนวนเท่ากับจำนวนลูกค้าทั้งหมดที่ผู้ใช้ระบุ เพื่อให้เข้าใจหลักการค้นหาคำตอบแบบสลับ 1 มากยิ่งขึ้น ซึ่งสามารถอธิบายได้ ดังตัวอย่างที่ 1

ตัวอย่างที่ 1 จากลูกค้าจำนวน 6 ราย สมมติว่า คำตอบปัจจุบันเป็น 2 – 6 – 3 – 1 – 4 - 5 ดังรูปที่ 4.1 ซึ่งแสดงตัวอย่างคำตอบปัจจุบัน



รูปที่ 4.1 ตัวอย่างคำตอบปัจจุบัน

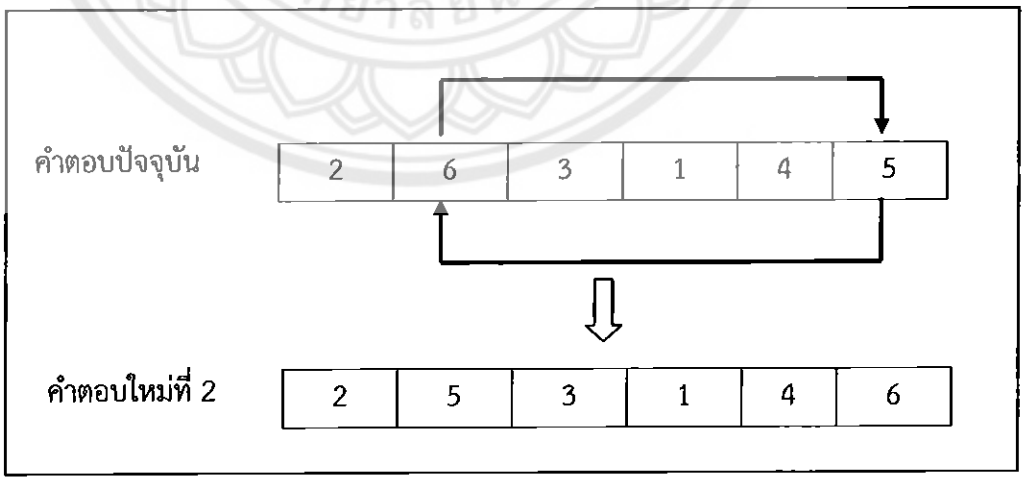
เมื่อได้คำตอบปัจจุบันแล้ว ทำการหาคำตอบใหม่ที่ 1 โดยการสุ่มตำแหน่งขึ้นมา 1 ตำแหน่ง เพื่อใช้ในการสลับที่กับตำแหน่งหลักตำแหน่งที่ 1 ซึ่งถ้ากำหนดให้ตำแหน่งที่สุ่มได้เป็นตำแหน่งที่ 3 ก็จะทำการสลับที่ระหว่างตำแหน่งที่ 1 กับตำแหน่งที่ 3 ดังรูป 4.2 ซึ่งแสดงการค้นหาคำตอบแบบสลับ 1 ของตำแหน่งหลักตำแหน่งที่ 1



รูปที่ 4.2 การค้นหาคำตอบแบบสลับ 1 ของตำแหน่งหลักตำแหน่งที่ 1

จากรูปที่ 4.2 หลังจากการสลับตำแหน่งหลักในตำแหน่งที่ 1 กับตำแหน่งที่ 3 ซึ่งเป็นตำแหน่งที่สุ่มมาได้แล้ว จะทำให้ได้คำตอบใหม่มา 1 คำตอบ คือ 3 – 6 – 2 – 1 – 4 - 5

จากนั้น ทำการหาคำตอบใหม่ที่ 2 โดยการสุ่มตำแหน่งขึ้นมา 1 ตำแหน่งเช่นเดียวกัน เพื่อใช้ในการสลับที่กับตำแหน่งหลักตำแหน่งที่ 2 ถ้ากำหนดให้ตำแหน่งที่สุ่มมาได้นั้นเป็นตำแหน่งที่ 6 จะทำการสลับที่ตำแหน่งที่ 2 กับตำแหน่งที่ 6 ดังรูป 4.3 ซึ่งแสดงการหาคำตอบแบบสลับ 1 ของตำแหน่งหลักที่ 2



รูปที่ 4.3 การหาคำตอบแบบสลับ 1 ของตำแหน่งหลักที่ 2

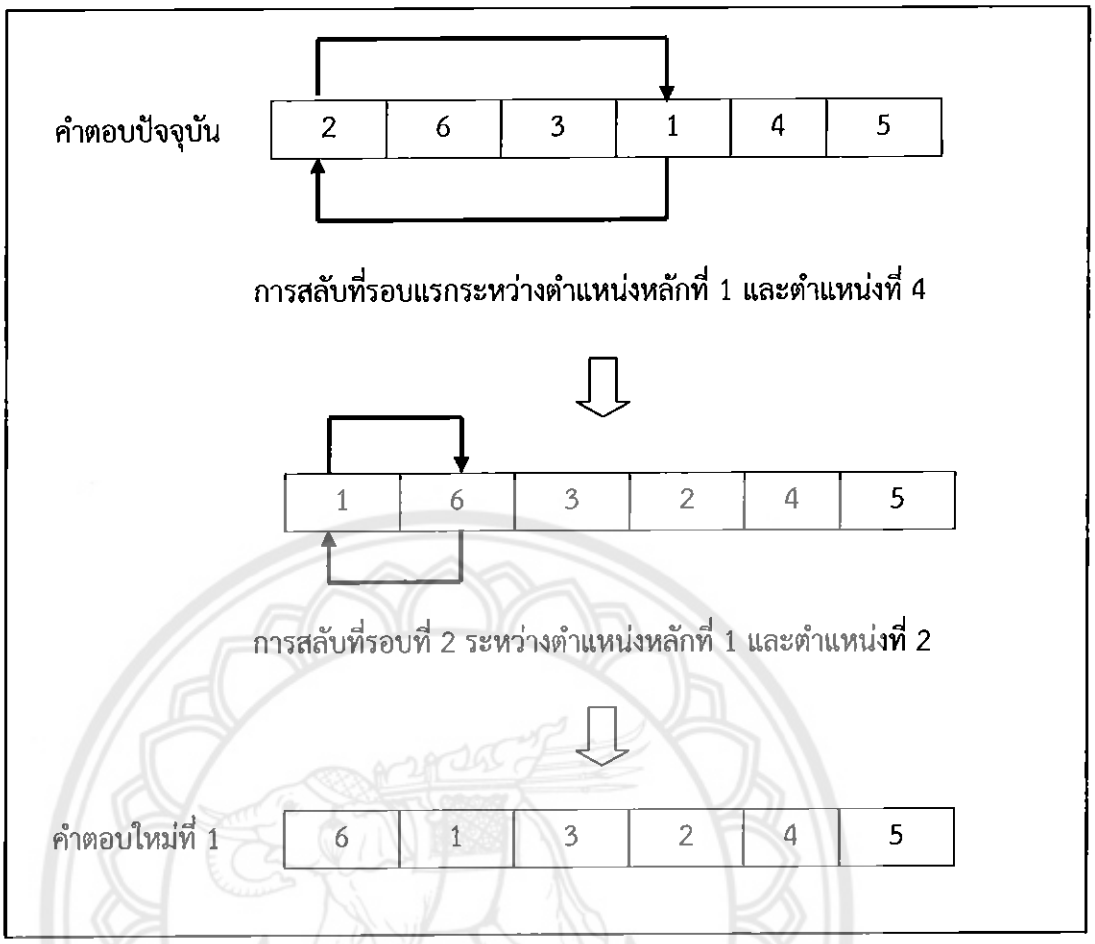
จากรูปที่ 4.3 หลักจากที่สลับที่ตำแหน่งหลักตำแหน่งที่ 2 กับตำแหน่งที่สุ่มมาได้ตำแหน่งที่ 6 แล้วจะทำให้ได้คำตอบใหม่อีก 1 คำตอบ เป็นคำตอบใหม่ที่ 2 คือ 2 - 5 – 3 – 1 – 4 - 6

จากตัวอย่างที่ 1 นี้ จะทำให้ได้คำตอบใหม่ทั้งหมด 6 คำตอบ ซึ่งจะมีจำนวนเท่ากับจำนวนลูกค้าตามที่โจทย์ระบุไว้ และสำหรับคำตอบใหม่ที่เหลือก็คือ คำตอบใหม่ที่ 3 ซึ่งจะมีตำแหน่งหลักเป็นตำแหน่งที่ 3 คำตอบใหม่ที่ 4 มีตำแหน่งหลักเป็นตำแหน่งที่ 4 คำตอบใหม่ที่ 5 มีตำแหน่งหลักเป็นตำแหน่งที่ 5 และคำตอบใหม่ที่ 6 จะมีตำแหน่งหลักเป็นตำแหน่งที่ 6

4.1.2.2 การค้นหาคำตอบแบบสลับ 2 (SWAP 2)

สำหรับการค้นหาคำตอบแบบสลับ 2 จะมีหลักการทำงานคล้ายคลึงกับการค้นหาคำตอบแบบสลับ 1 โดยการนำเอาตัวแทนคำตอบปัจจุบันมาใช้ในการค้นหาคำตอบแบบสลับ 2 จากนั้นทำการสุ่มตำแหน่งในคำตอบปัจจุบันขึ้นมา 1 ตำแหน่ง เพื่อใช้ในการสลับที่กับตำแหน่งหลัก แต่การค้นหาคำตอบแบบสลับ 2 จะแตกต่างจากการค้นหาคำตอบแบบสลับ 1 ตรงที่การค้นหาคำตอบแบบสลับ 2 เมื่อทำการสลับที่ระหว่างตำแหน่งที่สุ่มมาได้กับตำแหน่งหลักนั้นๆ 1 รอบแล้ว จะทำการสุ่มตำแหน่งในคำตอบปัจจุบันขึ้นมาอีก 1 ตำแหน่ง ซึ่งตำแหน่งใหม่ที่สุ่มมานั้นจะต้องไม่ซ้ำกับตำแหน่งที่สุ่มมาได้ก่อนหน้านี้และไม่ซ้ำกับตำแหน่งหลัก ถ้าตำแหน่งที่สุ่มมานั้นซ้ำกับตำแหน่งดังกล่าวจะต้องทำการสุ่มตำแหน่งใหม่ขึ้นมาอีกจนกระทั่งตำแหน่งนั้นไม่ซ้ำกัน สำหรับตำแหน่งที่สุ่มในรอบที่ 2 นี้ ก็จะนำไปใช้ในการสลับที่กับตำแหน่งหลักตำแหน่งเดิม ซึ่งตำแหน่งหลักนี้ก็จะให้เริ่มต้นด้วยตำแหน่งที่ 1 ในการหาคำตอบใหม่ที่ 1, ใช้ตำแหน่งหลักเป็นตำแหน่งที่ 2 สำหรับการหาคำตอบใหม่ที่ 2, ... , ใช้ตำแหน่งหลักที่ n ในการหาคำตอบใหม่ที่ n (เมื่อ n = จำนวนลูกค้าทั้งหมด) เช่นเดียวกับการค้นหาคำตอบแบบสลับ 1 ซึ่งอาจสรุปได้ว่า การค้นหาคำตอบแบบสลับ 2 นี้ คำตอบใหม่ที่ได้ใน 1 คำตอบ จะมีการสุ่มตำแหน่งจากคำตอบปัจจุบันเพื่อใช้ในสลับที่กับตำแหน่งหลักเดิมทั้งหมด 2 รอบ ซึ่งจำนวนคำตอบใหม่ที่ได้นั้นจะเท่ากับจำนวนของลูกค้าทั้งหมดเช่นเดียวกับการค้นหาคำตอบแบบสลับ 1 เพื่อให้เข้าใจหลักการการค้นหาคำตอบแบบสลับ 2 มากยิ่งขึ้น ซึ่งสามารถอธิบายได้ ดังตัวอย่างที่ 2

ตัวอย่างที่ 2 จากลูกค้าจำนวน 6 ราย สมมติว่า ให้คำตอบปัจจุบันเป็นคำตอบเดียวกันกับตัวอย่างที่ 1 คือ 2 – 6 – 3 – 1 – 4 – 5 ซึ่งจะนำคำตอบปัจจุบันนี้ไปใช้ในการค้นหาคำตอบแบบสลับ 2 โดยการสุ่มตำแหน่งขึ้นมาเพื่อสลับที่กับตำแหน่งหลักตำแหน่งที่ 1 ถ้ากำหนดให้ตำแหน่งที่สุ่มมาได้ในรอบแรกเป็นตำแหน่งที่ 4 และตำแหน่งที่สุ่มได้เพื่อมาสลับที่ในรอบที่ 2 เป็นตำแหน่งที่ 2 ซึ่งสามารถดูภาพประกอบของหลักการค้นหาคำตอบแบบสลับ 2 ได้ ดังรูป 4.4 ซึ่งจะแสดงการค้นหาคำตอบแบบสลับ 2 ของตำแหน่งหลักที่ 1



รูปที่ 4.4 การค้นหาคำตอบแบบสลับ 2 ของตำแหน่งหลักที่ 1

จากรูปที่ 4.4 เมื่อทำการสุ่มตำแหน่งจากคำตอบปัจจุบัน เพื่อใช้ในการสลับที่กับตำแหน่งหลักที่ 1 จำนวน 2 รอบ จะทำให้ได้คำตอบใหม่ 1 คำตอบ คือ 6 – 1 – 3 – 2 – 4 - 5

สำหรับการค้นหาคำตอบใหม่ในลำดับถัดไปก็จะทำในรูปแบบเดียวกันนี้ โดยคำตอบใหม่ที่ 2 จะมีตำแหน่งหลักเป็นตำแหน่งที่ 2, คำตอบใหม่ที่ 3 มีตำแหน่งหลักเป็นตำแหน่งที่ 3, คำตอบใหม่ที่ 4 มีตำแหน่งหลักเป็นตำแหน่งที่ 4, คำตอบใหม่ที่ 5 มีตำแหน่งหลักเป็นตำแหน่งที่ 5 และคำตอบใหม่ที่ 6 จะมีตำแหน่งหลักเป็นตำแหน่งที่ 6 ซึ่งจากตัวอย่างที่ 2 นี้ ทำให้ได้คำตอบใหม่ทั้งหมด 6 คำตอบ เช่นเดียวกับการค้นหาคำตอบแบบสลับ 1

4.1.3.3 การค้นหาคำตอบแบบเลื่อน (SLIDE)

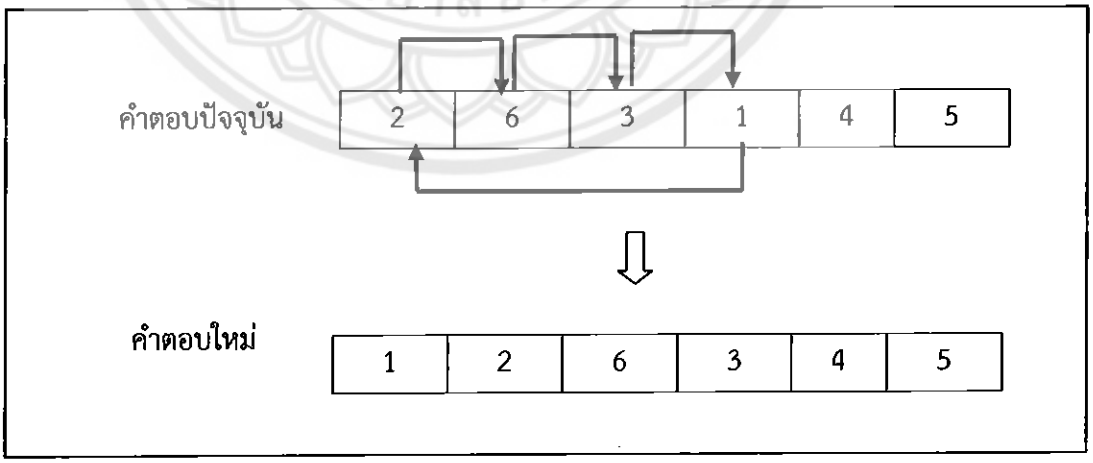
การค้นหาคำตอบแบบเลื่อน เป็นการค้นหาคำตอบโดยการนำเอาตัวแทนคำตอบปัจจุบันมาใช้ในการค้นหาคำตอบแบบเลื่อน เช่นเดียวกับการค้นหาคำตอบแบบสลับ 1 และการค้นหาคำตอบแบบสลับ 2 โดยทำการสุ่มตำแหน่งขึ้นมา 1 ตำแหน่ง โดยตำแหน่งที่สุ่มมานั้นจะต้องไม่ซ้ำกับตำแหน่งหลัก ถ้าซ้ำกันจะต้องทำการสุ่มตำแหน่งใหม่จนกระทั่งไม่ซ้ำกับตำแหน่งหลักนั้น สำหรับวิธีนี้จะมีหลักการคล้ายๆ กับการค้นหาคำตอบแบบสลับ 1 แต่จะแตกต่างกันที่วิธีการสลับตำแหน่ง สำหรับการค้นหาคำตอบแบบเลื่อนนี้ เมื่อสุ่มตำแหน่งในคำตอบปัจจุบันได้แล้วก็จะทำการเลื่อน

ตำแหน่ง โดยการนำค่าของตำแหน่งที่สุ่มได้ไปแทนที่ในค่าของตำแหน่งหลัก แล้วให้เลื่อนค่าของตำแหน่งหลักนั้นไปยังตำแหน่งถัดไปทางขวามือ โดยจะทำการเลื่อนตำแหน่งไปเรื่อยๆ จนกระทั่งถึงตำแหน่งที่สุ่มมาได้ ก็จะทำให้ได้คำตอบใหม่ขึ้นมา 1 คำตอบ ซึ่งสำหรับรูปแบบการเลื่อนที่กล่าวมานี้จะใช้ในกรณีที่ตำแหน่งที่สุ่มมาได้นั้นมีค่ามากกว่าตำแหน่งหลัก เช่น ตำแหน่งหลักเป็นตำแหน่งที่ 1 และตำแหน่งที่สุ่มมาได้นั้นเป็นตำแหน่งที่ 4 หรือตำแหน่งหลักเป็นตำแหน่งที่ 2 และตำแหน่งที่สุ่มมานั้นเป็นตำแหน่งที่ 6 เป็นต้น แต่ถ้าตำแหน่งที่สุ่มมานั้นน้อยกว่าตำแหน่งหลัก เช่น ถ้าตำแหน่งหลักเป็นตำแหน่งที่ 4 แต่ตำแหน่งที่สุ่มได้นั้นเป็นตำแหน่งที่ 2 ก็จะใช้หลักการในการเลื่อนที่เหมือนกัน เพียงแต่กรณีนี้จะเลื่อนตำแหน่งไปยังซ้ายมือ โดยจะนำค่าจากตำแหน่งที่สุ่มได้ไปแทนที่ค่าของตำแหน่งหลักเช่นเดียวกับกรณีแรก จากนั้นก็ทำการเลื่อนค่าจากตำแหน่งหลักมายังตำแหน่งก่อนหน้าทางซ้ายมือ ซึ่งจะทำการเลื่อนตำแหน่งรูปแบบนี้ไปเรื่อยๆ จนกระทั่งถึงตำแหน่งที่สุ่มได้ เพื่อให้เข้าใจหลักการการค้นหาคำตอบแบบเลื่อนมากยิ่งขึ้น ซึ่งสามารถอธิบายได้ ดังตัวอย่างที่ 3

ตัวอย่างที่ 3 จากลูกค้าจำนวน 6 ราย สมมติว่า ให้คำตอบปัจจุบันเป็นคำตอบเดิมจากตัวอย่างที่ 1 คือ 2 – 6 – 3 – 1 – 4 – 5 ซึ่งจะนำคำตอบปัจจุบันนี้ไปใช้ในการค้นหาคำตอบแบบเลื่อน ซึ่งสามารถดูภาพประกอบของหลักการค้นหาคำตอบแบบเลื่อนได้ ดังรูป 4.5 ซึ่งแสดงการค้นหาคำตอบแบบเลื่อน กรณีที่ตำแหน่งที่สุ่มมานั้นมีค่ามากกว่าตำแหน่งหลักและ รูปที่ 4.6 ซึ่งแสดงการค้าหาคำตอบแบบเลื่อน กรณีที่ตำแหน่งที่สุ่มมานั้นมีค่าน้อยกว่าตำแหน่งหลัก

กรณีที่ตำแหน่งที่สุ่มมานั้นมีค่ามากกว่าตำแหน่งหลัก

โดยเริ่มจากตำแหน่งหลักตำแหน่งที่ 1 ซึ่งกำหนดให้ตำแหน่งที่สุ่มได้นั้นเป็นตำแหน่งที่ 4

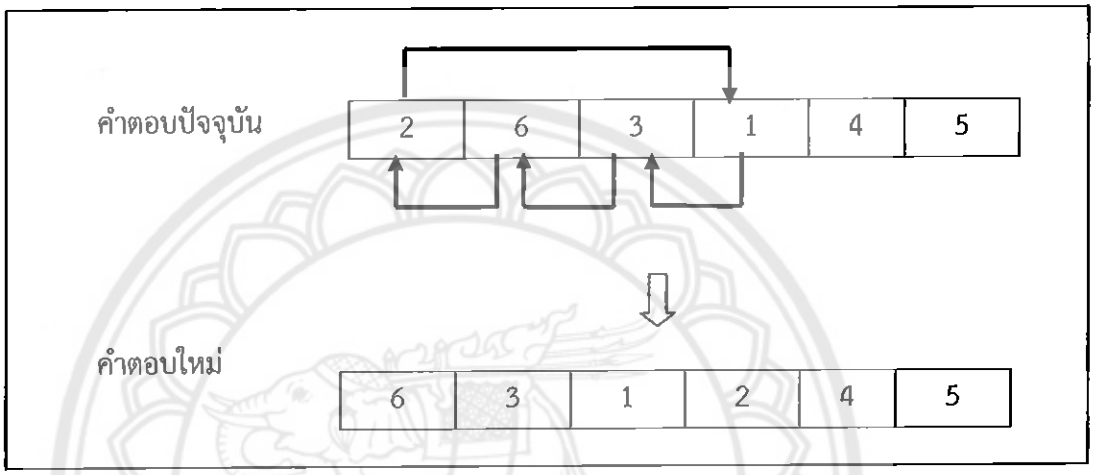


รูปที่ 4.5 การค้นหาคำตอบแบบเลื่อน กรณีที่ตำแหน่งที่สุ่มมานั้นมีค่ามากกว่าตำแหน่งหลัก

จากรูปที่ 4.5 ในการค้นหาคำตอบแบบเลื่อนจากตัวอย่างข้างต้นนี้ ในกรณีที่ตำแหน่งที่สุ่มมานั้นมีค่ามากกว่าตำแหน่งหลัก โดยที่ตำแหน่งหลักเป็นตำแหน่งที่ 1 และตำแหน่งที่สุ่มมาได้นั้นเป็นตำแหน่งที่ 4 ซึ่งจะทำให้ได้คำตอบใหม่ 1 คำตอบ คือ 1 – 2 – 6 – 3 – 4 - 5

กรณีที่ตำแหน่งที่สุ่มมานั้นมีค่าน้อยกว่าตำแหน่งหลัก

ถ้าตำแหน่งหลักเป็นตำแหน่งที่ 4 และกำหนดให้ตำแหน่งที่สุ่มได้นั้นเป็นตำแหน่งที่ 1



รูปที่ 4.6 การค้าหาคำตอบแบบเลื่อน กรณีที่ตำแหน่งที่สุ่มมานั้นมีค่าน้อยกว่าตำแหน่งหลัก

จากรูปที่ 4.6 สำหรับการค้นหาคำตอบแบบเลื่อนนี้ ในกรณีที่ตำแหน่งที่สุ่มมานั้นมีค่าน้อยกว่าตำแหน่งหลัก โดยที่ตำแหน่งหลักเป็นตำแหน่งที่ 4 และตำแหน่งที่สุ่มมาได้นั้นเป็นตำแหน่งที่ 1 ซึ่งจะทำให้ได้คำตอบใหม่ 1 คำตอบ คือ 6 – 3 – 1 – 2 – 4 - 5

สำหรับการค้นหาคำตอบแบบเลื่อนจากตัวอย่างนี้จะทำให้ได้คำตอบใหม่ทั้งหมด 6 คำตอบ เช่นเดียวกันกับการค้นหาคำตอบแบบสลับ 1 และการค้นหาคำตอบแบบสลับ 2

4.1.3 กระบวนการจัดเรียงรอบรถ

เมื่อเราทำการค้นหาคำตอบปัจจุบันจากทั้ง 3 รูปแบบ ดังที่ได้กล่าวไว้ข้างต้นเรียบร้อยแล้ว จากนั้นก็จะเข้าสู่กระบวนการจัดเรียงรอบรถ ซึ่งกระบวนการนี้จะเป็นกระบวนการจัดการเพื่อที่จะทำการขนส่งสินค้าให้กับลูกค้าแต่ละรายภายใต้ขีดจำกัดความจุของยานพาหนะ โดยจะทำให้ทราบว่ารถแต่ละคันจะต้องไปส่งสินค้าให้กับลูกค้ารายไหนบ้าง ซึ่งรถแต่ละคันนั้นจะมีความจุเท่ากัน โดยหลักในการจัดเรียงรอบรถนี้ จะเริ่มจากการรวมความจุของสินค้าที่ลูกค้าต้องการในตำแหน่งที่ 1 กับความจุของสินค้าที่ลูกค้าต้องการในตำแหน่งที่ 2 แล้วพิจารณาว่าความจุของสินค้าที่ลูกค้าทั้ง 2 ตำแหน่งต้องการนั้น มีความจุเกินความจุสูงสุดของรถคันที่ 1 หรือไม่ ถ้าผลรวมความจุนั้นไม่เกินความจุสูงสุดของรถก็จะทำการรวมความจุที่ลูกค้าต้องการในตำแหน่งถัดไปเรื่อยๆ แต่ถ้าผลรวมของความจุที่ลูกค้า

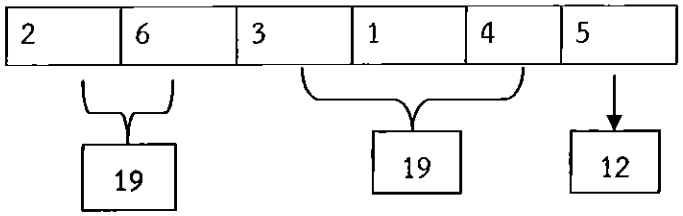
ต้องการนั้นเกินความจุสูงสุดของรถแล้ว ก็จะนำความจุที่ลูกค้าต้องการในตำแหน่งที่ทำให้ผลรวมของความจุสินค้าเกินความจุของรถ ไปเริ่มต้นรวมความจุของสินค้าในรถคันถัดไป ถ้ากรณีที่ผลรวมของความจุสินค้านั้นมีค่าเท่ากับความจุสูงสุดของรถพอดี ความจุของสินค้าที่ลูกค้าต้องการในตำแหน่งถัดไปก็จะไปเริ่มต้นรวมความจุสินค้าใหม่ในรถคันถัดไปเช่นเดียวกัน แต่ถ้ากรณีที่ความต้องการสินค้าของลูกค้าทั้งหมดนั้นมีปริมาณมากกว่าจำนวนรถก็จะต้องการเสียค่าปรับโดยอัตโนมัติ ซึ่งจะทำในรูปแบบนี้ไปเรื่อยๆ จนครบตามจำนวนลูกค้าทั้งหมด โดยทุกๆ ครั้งที่ความจุของรถคันนั้นๆ ไม่สามารถบรรจุสินค้าเพิ่มได้แล้ว ก็จะใส่ 0 หลังจากตำแหน่งลูกค้ารายสุดท้ายที่รถคันนั้นจะต้องไปส่งสินค้า ซึ่งจำนวนของ 0 ทั้งหมด จะมีจำนวนน้อยกว่าเท่ากับจำนวนรถที่ผู้ใช้ระบุไว้ ในการกรอกข้อมูลเบื้องต้น เพื่อให้เข้าใจกระบวนการจัดเรียงรอบรถมากยิ่งขึ้น ซึ่งสามารถอธิบายได้ ดังตัวอย่างที่ 4

ตัวอย่างที่ 4 สมมติให้คำตอบปัจจุบันเป็น 2 – 6 – 3 – 1 – 4 – 5 และกำหนดให้มีรถทั้งหมด 3 คัน โดยรถแต่ละมีความจุเท่ากัน คือ 20 ตัน มีลูกค้าที่ต้องไปขนส่งสินค้าทั้งหมด 6 ราย โดยลูกค้าแต่ละรายมีความต้องการสินค้า ดังตารางที่ 4.1 ซึ่งแสดงตัวอย่างความต้องการของลูกค้าทั้ง 6 ราย ดังรูปที่ 4.2 แสดงกระบวนการจัดเรียงรอบรถ

ตารางที่ 4.1 แสดงตัวอย่างความต้องการของลูกค้าทั้ง 6 ราย

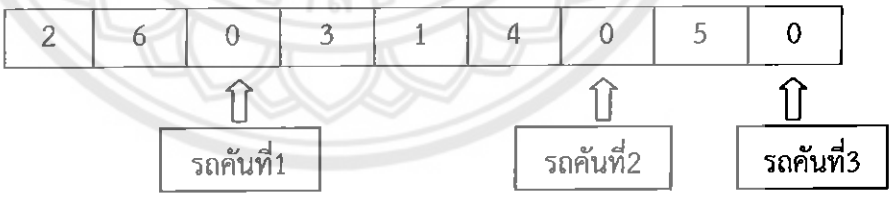
ลูกค้ารายที่	ความต้องการสินค้าของลูกค้าแต่ละราย (ตัน)
1	8
2	7
3	6
4	5
5	9
6	12

นำตำแหน่งของคำตอบปัจจุบันมาทำการรวมความจุของสินค้าที่ลูกค้าแต่ละรายต้องการ ซึ่งผลรวมความจุของสินค้าในแต่ละตำแหน่งจะต้องน้อยกว่า หรือเท่ากับความจุของยานพาหนะ



เมื่อทำการการจัดเรียงลำดับการขนส่งของรถแต่ละคันเรียบร้อยแล้ว จะได้ผลลัพธ์ ดังนี้

จากนั้นทำการใส่ 0 คั่น ระหว่างรถแต่ละคัน ซึ่งจำนวนของ 0 นั้น จะเท่ากับจำนวนรถที่ผู้ใช้ได้ระบุไว้ในการกรอกข้อมูลเบื้องต้น ดังนี้



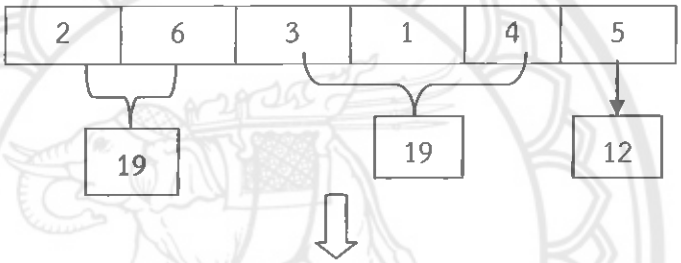
รูปที่ 4.7 แสดงกระบวนการจัดเรียงรอบรถ

จากรูปที่ 4.7 เมื่อทำการจัดเรียงรอบรถของคำตอบปัจจุบันแล้ว จะทำให้ทราบว่า รถคันที่ 1 จะต้องเริ่มต้นส่งสินค้าให้กับลูกค้ารายที่ 2 ก่อน ซึ่งลูกค้ารายที่ 2 มีความต้องการสินค้าเท่ากับ 7 ตัน จากนั้นก็ไปส่งสินค้าให้กับลูกค้ารายที่ 6 เป็นลำดับถัดไป ซึ่งลูกค้ารายที่ 6 นี้ มีความต้องการสินค้าเท่ากับ 12 ตัน ซึ่งผลรวมความจุของลูกค้าทั้งสองรายเท่ากับ 19 ตัน ซึ่งยังไม่เกินความจุสูงสุดของรถที่ระบุไว้ คือ 20 ตัน แต่ก็ไม่สามารถไปส่งสินค้าให้กับลูกค้ารายที่ 3 ต่อได้ เพราะเนื่องจากถ้ารวมความต้องการสินค้าของลูกค้ารายที่ 3 ด้วย จะทำให้ความจุรวมสินค้านั้นเกินกว่าความจุสูงสุดของ

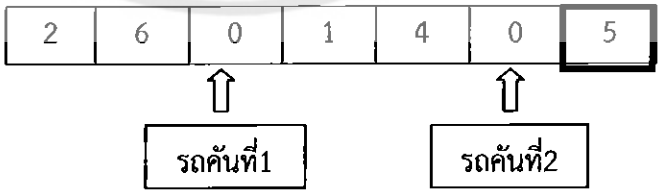
รถ 1 คัน ดังนั้น ลูกค้ารายที่ 3 จึงได้รับสินค้าจากรถคันที่ 2 รวมทั้งลูกค้ารายที่ 1 และลูกค้ารายที่ 4 ด้วย ซึ่งเมื่อรวมความจุของลูกค้าทั้ง 3 รายแล้ว ความจุของสินค้ายังไม่เกินกว่าความจุของรถ 1 คัน และลูกค้ารายที่ 5 ก็จะได้รับสินค้าจากรถคันที่ 3

ตัวอย่างที่ 5 ถ้ากรณีที่ความต้องการสินค้าของลูกค้าทั้งหมดนั้นมีปริมาณมากกว่าจำนวนรถ จากตัวอย่างโจทย์ที่ 4 ถ้ากำหนดให้มีรถที่สามารถไปส่งสินค้าให้กับลูกค้าได้เพียง 2 คัน ซึ่งในกรณีนี้ความต้องการสินค้าของลูกค้าทั้งหมดนั้นมีปริมาณมากกว่าจำนวนรถที่มีอยู่ ก็จะกำหนดให้ต้องเสียค่าปรับโดยอัตโนมัติ

ทำหลักการเดียวกับ ตัวอย่างที่ 4 โดยการนำตำแหน่งของคำตอบปัจจุบันมาทำการรวมความจุของสินค้าที่ลูกค้าแต่ละรายต้องการ



เมื่อทำการจัดเรียงลำดับการขนส่งของรถแต่ละคันเรียบร้อยแล้ว จากนั้นทำการใส่ 0 คั่น ระหว่างรถแต่ละคัน ซึ่งจำนวนของ 0 นั้น จะเท่ากับจำนวนรถที่ผู้ใช้ได้ระบุไว้ในการกรอกข้อมูลเบื้องต้น จะได้ผลลัพธ์ ดังนี้ ความต้องการสินค้าของลูกค้าทั้งหมดนั้นมีปริมาณมากกว่าความจุรถที่มีอยู่



ซึ่งจะเห็นได้ว่าจากตารางจากจัดเรียงรถข้างบน จะมีความต้องการของลูกค้ารายที่ 5 เกินจากความจุของรถที่มีอยู่ แต่ในกรณีนี้จะถูกกำหนดให้ต้องมีการเสียค่าปรับโดยอัตโนมัติ

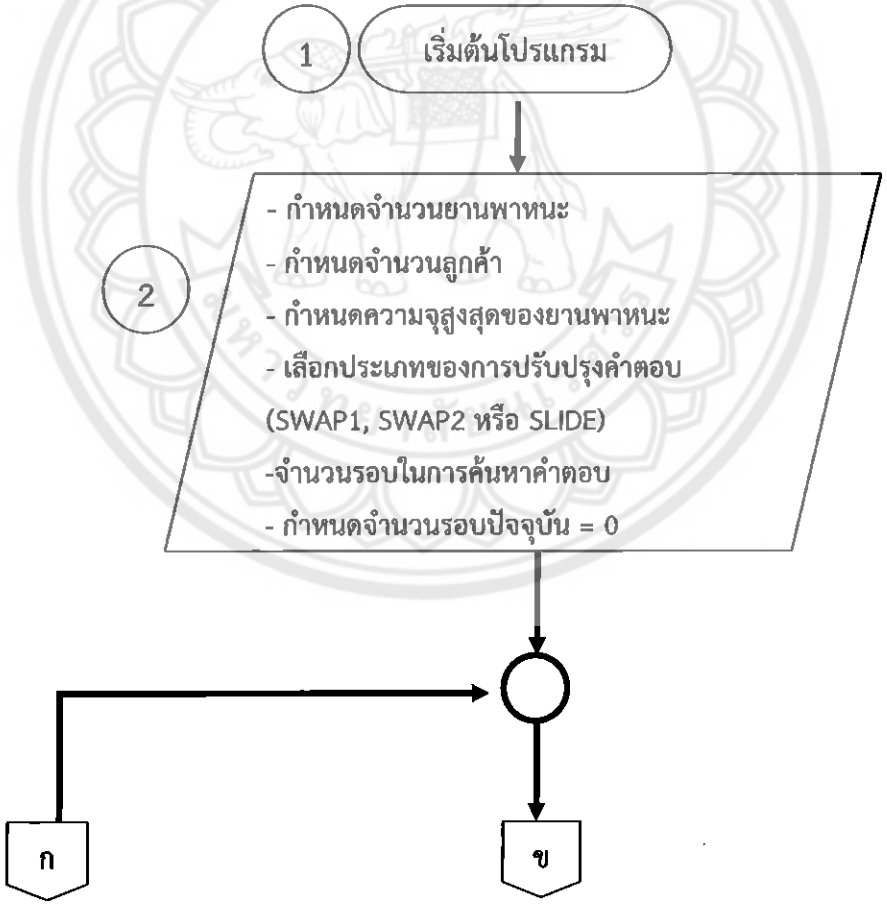
รูปที่ 4.8 แสดงกระบวนการจัดเรียงรอบรถสินค้ามีปริมาณมากกว่าจำนวนรถ

4.1.4 การออกแบบเมตาฮิวริสติก

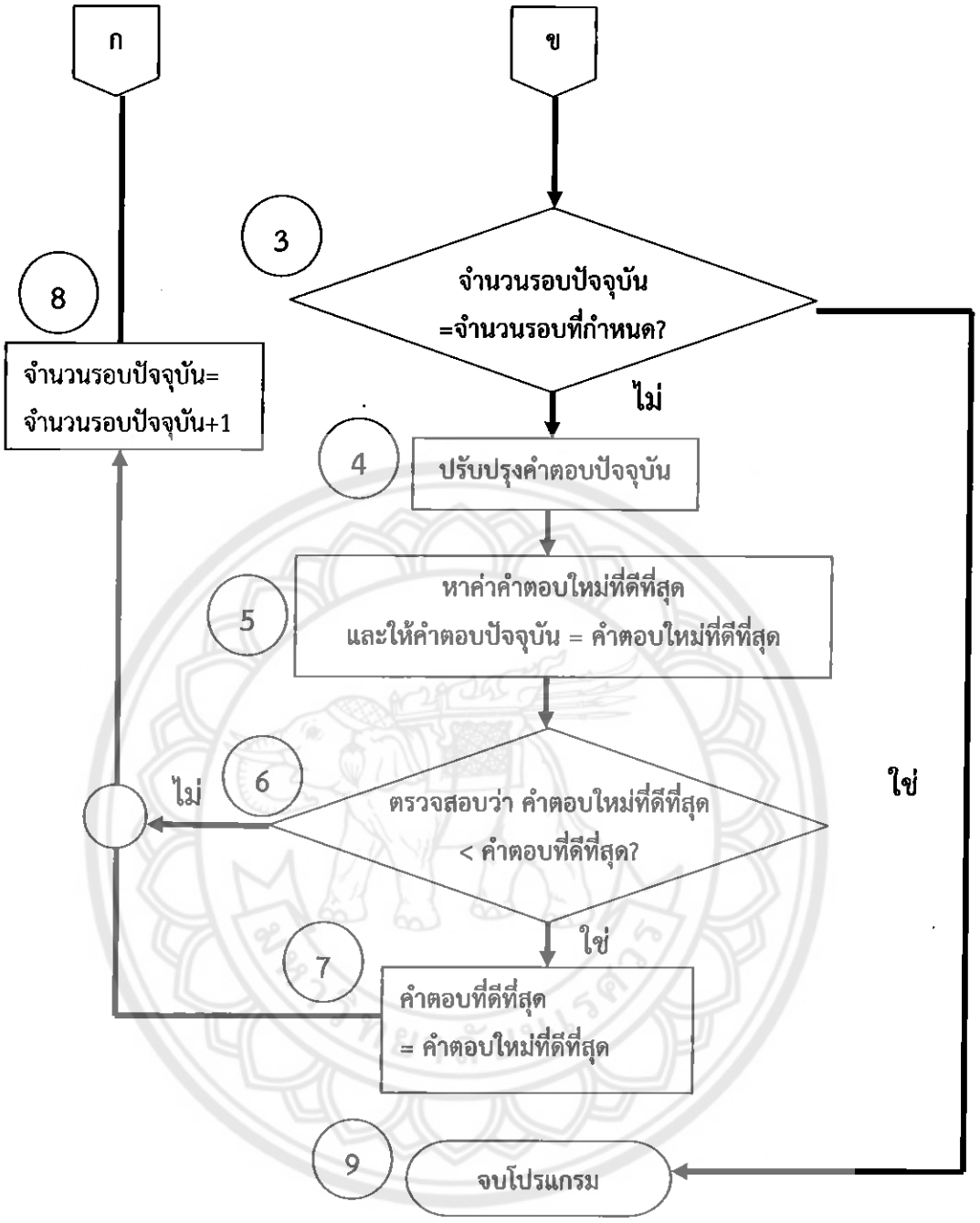
สำหรับการออกแบบเมตาฮิวริสติกเพื่อใช้ในการค้นหาคำตอบที่ดีที่สุดนั้น จะมีด้วยกันทั้งหมด 3 รูปแบบ คือ การออกแบบการทำงานของ Local Search Type 1 (LS1) การออกแบบการทำงานของ Local Search Type 2 (LS2) และการออกแบบการทำงานของ Iterated Local Search (ILS)

4.1.4.1 การออกแบบการทำงานของ Local Search Type 1

การทำงานของ Local Search Type 1 จะเป็นการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ โดยใช้การค้นหาคำตอบเฉพาะที่ (Local Search) ซึ่งจะนำคำตอบใหม่ที่มีระยะทางรวมที่สั้นที่สุดในแต่ละรอบมาแทนที่คำตอบปัจจุบัน สำหรับวิธีการหาคำตอบก็จะมีทั้งหมด 3 วิธี คือ วิธีการหาคำตอบแบบสลับ 1 วิธีการหาคำตอบแบบสลับ 2 และวิธีการหาคำตอบแบบเลื่อน ซึ่งจากรูปที่ 4.9 นี้ จะแสดงหลักการการทำงานของ Local Search Type 1



รูปที่ 4.9 การทำงานของ Local Search Type 1



รูปที่ 4.9 (ต่อ) การทำงานของ Local Search Type 1

จากรูปที่ 4.9 จะเห็นว่ามีการทำงานของ Local Search Type 1 ตามขั้นตอนดังนี้

ขั้นตอนที่ 1 เริ่มต้นโปรแกรมจากการเปิดหน้าต่างโปรแกรมขึ้นมา

ขั้นตอนที่ 2 กำหนดจำนวนยานพาหนะ จำนวนลูกค้า ความจุสูงสุดของยานพาหนะ เลือกประเภทของการปรับปรุงโดยมีให้เลือก 3 ประเภท คือ การค้นหาคำตอบแบบสลับ 1 (Swap1) การค้นหาคำตอบแบบสลับ 2 (Swap 2) และการค้นหาคำตอบแบบเลื่อน (Slide) และ

กำหนดจำนวนรอบในการค้นหาคำตอบ ซึ่งเป็นการทำงานของโปรแกรมว่าจะทำการค้นหาคำตอบกี่รอบ โปรแกรมจึงจะหยุดทำงาน

ขั้นตอนที่ 3 ตรวจสอบว่าจำนวนรอบปัจจุบันเท่ากับจำนวนรอบการค้นหาคำตอบที่กำหนดหรือไม่ ถ้าไม่จะทำต่อในขั้นตอนที่ 4 แต่ถ้าใช่ก็จะหยุดโปรแกรมการทำงาน

ขั้นตอนที่ 4 ทำการปรับปรุงคำตอบปัจจุบัน

ขั้นตอนที่ 5 เมื่อได้ค่าคำตอบใหม่ที่ดีที่สุด จะให้คำตอบใหม่ที่ดีที่สุดนั้นมีค่าเท่ากับคำตอบปัจจุบัน โดยไม่สนใจว่าจะมีค่ามากกว่าคำตอบปัจจุบันหรือไม่ และนำคำตอบใหม่ที่ดีที่สุดนั้นไปเปรียบเทียบกับคำตอบที่ดีที่สุด

ขั้นตอนที่ 6 ตรวจสอบว่าคำตอบใหม่ที่ดีที่สุดนั้นมีค่าน้อยกว่าคำตอบที่ดีที่สุดหรือไม่

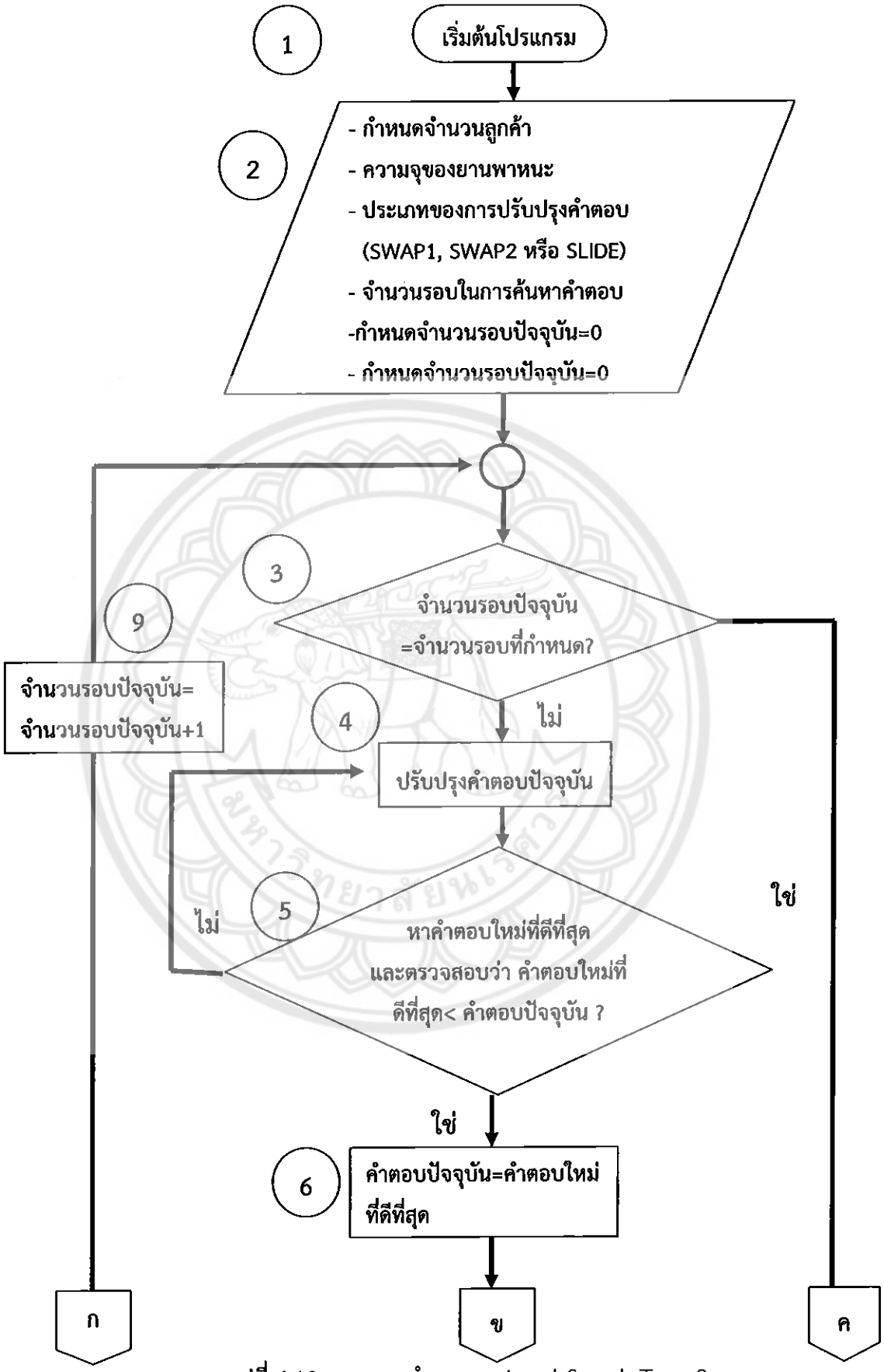
ขั้นตอนที่ 7 ถ้าคำตอบใหม่ที่ดีที่สุดมีค่าน้อยกว่าคำตอบที่ดีที่สุด จะให้ค่าคำตอบใหม่ไปแทนที่คำตอบที่ดีที่สุดนั้น แต่ถ้าไม่ดีกว่าก็จะให้จำนวนรอบปัจจุบันเท่ากับจำนวนรอบปัจจุบันบวก 1 และทำการปรับปรุงคำตอบต่อไป

ขั้นตอนที่ 8 ให้จำนวนรอบปัจจุบันเท่ากับจำนวนรอบปัจจุบันบวก 1

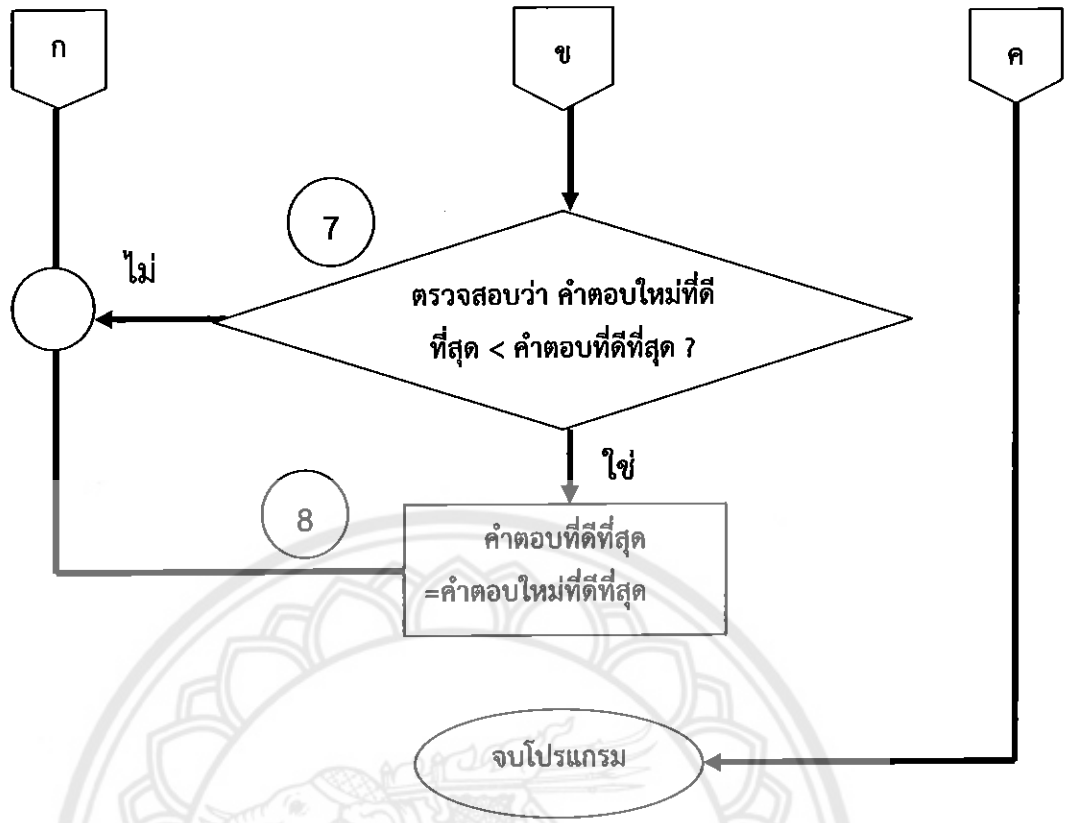
ขั้นตอนที่ 9 เมื่อจำนวนรอบปัจจุบันเท่ากับจำนวนรอบที่ผู้ใช้กำหนดแล้ว โปรแกรมจะหยุดการทำงาน

4.1.4.2 การออกแบบการทำงานของ Local Search Type 2

การทำงานของ Local Search Type 2 จะเป็นการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ โดยใช้การค้นหาคำตอบเฉพาะที่ (Local Search) ซึ่งคำตอบใหม่ที่ดีที่สุดที่จะมาแทนที่คำตอบปัจจุบันได้นั้นจะต้องมีค่าน้อยกว่าคำตอบปัจจุบัน และสำหรับวิธีการหาคำตอบก็จะมีทั้งหมด 3 วิธี คือ วิธีการหาคำตอบแบบสลับ 1 วิธีการหาคำตอบแบบสลับ 2 และวิธีการหาคำตอบแบบเลื่อน ซึ่งจากรูปที่ 4.10 นี้ จะแสดงหลักการการทำงานของ Local Search Type 2



รูปที่ 4.10 แสดงการทำงานของ Local Search Type 2



รูปที่ 4.10 (ต่อ) แสดงการทำงานของ Local Search Type 2

จากรูปที่ 4.10 จะเห็นว่ามีการทำงานของ Local Search Type 2 ตามขั้นตอนดังนี้

ขั้นตอนที่ 1 เริ่มต้นโปรแกรมโดยการเปิดหน้าต่างโปรแกรมขึ้นมา

ขั้นตอนที่ 2 กำหนดจำนวนยานพาหนะ จำนวนลูกค้า ความจุสูงสุดของยานพาหนะ เลือกประเภทของการปรับปรุงโดยมีให้เลือก 3 ประเภท คือ การค้นหาคำตอบแบบสลับ 1 (Swap 1) การค้นหาคำตอบแบบสลับ 2 (Swap 2) และการค้นหาคำตอบแบบเลื่อน (Slide) กำหนดจำนวนรอบในการค้นหาคำตอบ ซึ่งเป็นการทำงานของโปรแกรมว่าจะทำการค้นหาคำตอบกี่รอบ โปรแกรมจึงจะหยุดทำงาน และกำหนดจำนวนรอบปัจจุบันเท่ากับศูนย์

ขั้นตอนที่ 3 ตรวจสอบว่าจำนวนรอบปัจจุบันเท่ากับจำนวนรอบการค้นหาคำตอบที่กำหนดหรือไม่ ถ้าไม่จะทำต่อในขั้นตอนที่ 4 แต่ถ้าใช่ก็จะหยุดโปรแกรมการทำงาน

ขั้นตอนที่ 4 ทำการปรับปรุงคำตอบปัจจุบัน

ขั้นตอนที่ 5 เมื่อได้คำตอบใหม่ที่ดีที่สุด จะตรวจสอบว่าคำตอบใหม่ มีค่าน้อยกว่าปัจจุบันหรือไม่ ถ้าไม่น้อยกว่าก็จะให้จำนวนรอบปัจจุบันเท่ากับจำนวนรอบปัจจุบันบวก 1 และทำการปรับปรุงคำตอบต่อไป แต่ถ้าน้อยกว่าก็จะให้ทำในขั้นตอนที่ 6

ขั้นตอนที่ 6 ถ้าคำตอบใหม่ที่ดีที่สุดมีค่าน้อยกว่าคำตอบปัจจุบัน จะให้ค่าคำตอบใหม่นั้นไปแทนที่คำตอบปัจจุบัน

ขั้นตอนที่ 7 ตรวจสอบว่าคำตอบใหม่ที่ดีที่สุดมีค่าน้อยกว่าคำตอบที่ดีที่สุดหรือไม่ ถ้าน้อยกว่าก็จะให้ทำในขั้นตอนที่ 8 แต่ถ้าไม่น้อยกว่าก็จะให้ปรับปรุงคำตอบในรอบต่อไป

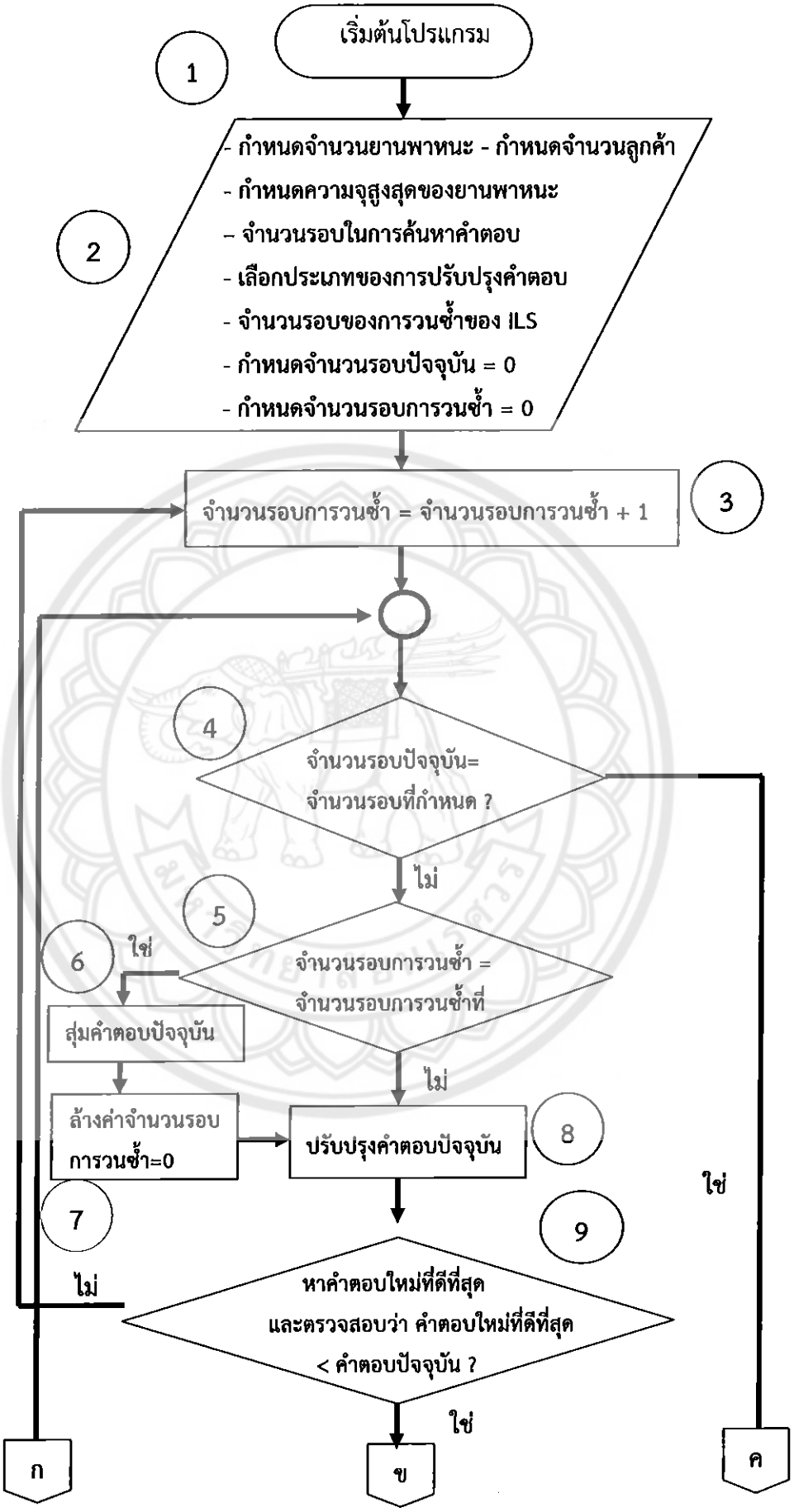
ขั้นตอนที่ 8 ถ้าคำตอบใหม่ที่ดีที่สุดมีค่าน้อยกว่าคำตอบที่ดีที่สุด ก็จะให้คำตอบใหม่ที่ดีที่สุดนั้นไปแทนที่คำตอบที่ดีที่สุด

ขั้นตอนที่ 9 ให้จำนวนรอบปัจจุบันเท่ากับจำนวนรอบปัจจุบันบวก 1

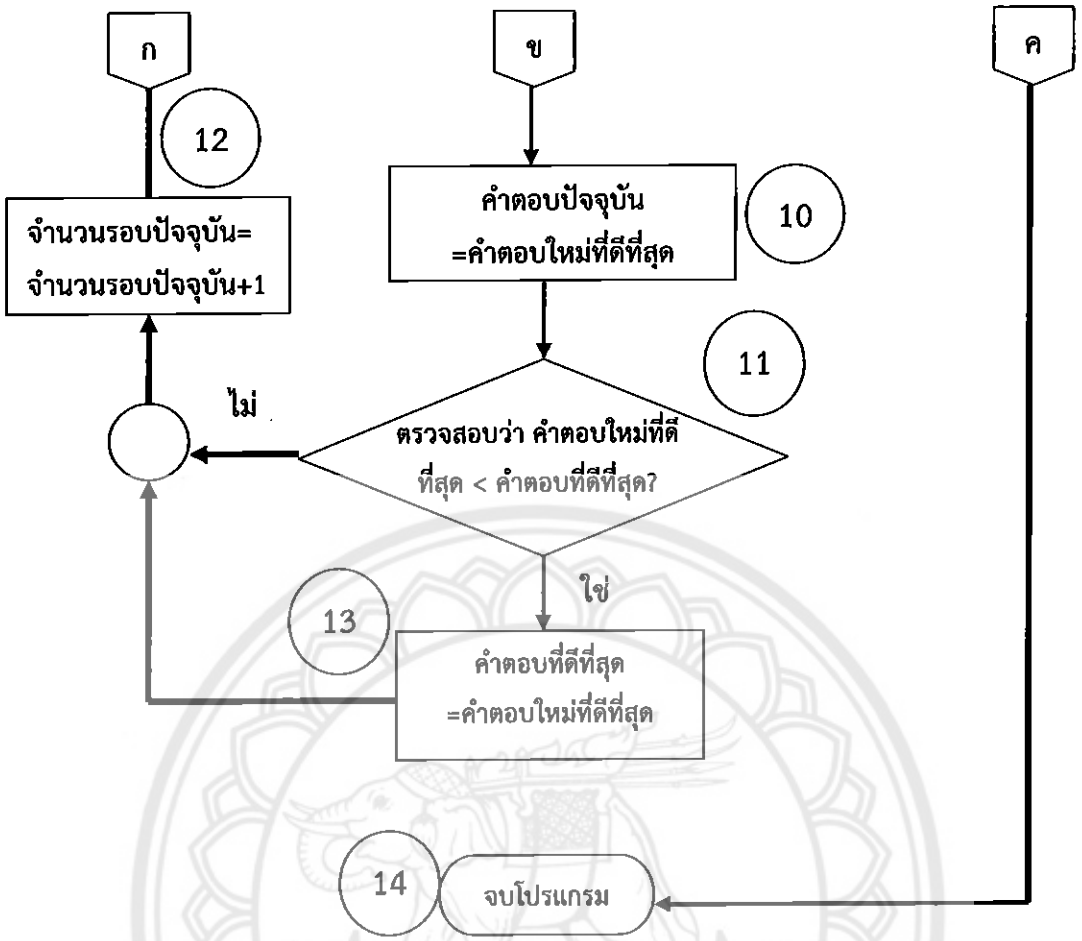
ขั้นตอนที่ 10 ถ้าจำนวนรอบปัจจุบันเท่ากับจำนวนรอบที่ผู้ใช้กำหนดแล้ว โปรแกรมจะหยุดการทำงาน

4.1.4.3 การออกแบบการทำงานของ Iterated Local Search

การทำงานของ Iterated Local Search จะเป็นการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุ โดยใช้การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ (Iterated Local Search) สำหรับการทำงานของ Iterated Local Search จะมีการกำหนดจำนวนรอบของการวนซ้ำของคำตอบปัจจุบัน ในกรณีที่ไม่พบคำตอบค่าคำตอบใหม่ที่ดีกว่าค่าคำตอบปัจจุบันแล้วจะให้วนซ้ำคำตอบปัจจุบันกี่รอบจึงจะทำการสุ่มคำตอบปัจจุบันขึ้นมาใหม่ และสำหรับวิธีการหาคำตอบก็จะมีทั้งหมด 3 วิธี คือ วิธีการหาคำตอบแบบสลับ 1 วิธีการหาคำตอบแบบสลับ 2 และวิธีการหาคำตอบแบบเลื่อน โดยมีขั้นตอนการหาคำตอบ ดังรูปที่ 4.11 ซึ่งแสดงการทำงานของ Iterated Local Search



รูปที่ 4.11 แสดงการทำงานของ Iterated Local Search



รูปที่ 4.11 (ต่อ) แสดงการทำงานของ Iterated Local Search

จากรูปที่ 4.10 จะเห็นว่ามีการการทำงานของ Iterated Local Search ตามขั้นตอนดังนี้

ขั้นตอนที่ 1 เริ่มต้นโปรแกรมโดยการเปิดหน้าต่างโปรแกรมขึ้นมา

ขั้นตอนที่ 2 กำหนดจำนวนยานพาหนะ จำนวนลูกค้า ความจุสูงสุดของยานพาหนะ เลือกประเภทของการปรับปรุงโดยมีให้เลือก 3 ประเภท คือ การค้นหาคำตอบแบบสลับ 1 (Swap 1) การค้นหาคำตอบแบบสลับ 2 (Swap 2) และการค้นหาคำตอบแบบเลื่อน (Slide) และกำหนดจำนวนรอบในการค้นหาคำตอบ ซึ่งเป็นการทำงานของโปรแกรมว่าจะทำการค้นหาคำตอบกี่รอบ โปรแกรมจึงจะหยุดทำงาน และกำหนดจำนวนรอบปัจจุบันและจำนวนรอบการวนซ้ำเท่ากับศูนย์

ขั้นตอนที่ 3 ให้จำนวนรอบการวนซ้ำเท่ากับจำนวนรอบวนซ้ำบวก 1

ขั้นตอนที่ 4 ตรวจสอบว่าจำนวนรอบปัจจุบันเท่ากับจำนวนรอบการค้นหาคำตอบที่กำหนดหรือไม่ ถ้าไม่จะทำต่อในขั้นตอนที่ 5 แต่ถ้าใช่ก็จะหยุดโปรแกรมการทำงาน

ขั้นตอนที่ 5 ตรวจสอบว่าจำนวนรอบการวนซ้ำเท่ากับจำนวนรอบการวนซ้ำที่กำหนดหรือไม่ ถ้าไม่จะทำต่อในขั้นตอนที่ 8 แต่ถ้าใช่ก็จะทำในขั้นตอนที่ 6

ขั้นตอนที่ 6 ทำการสุ่มคำตอบปัจจุบันใหม่

ขั้นตอนที่ 7 เมื่อทำการสุ่มคำตอบปัจจุบันใหม่แล้ว ให้ล้างค่าจำนวนรอบการวนซ้ำเป็นศูนย์

ขั้นตอนที่ 8 ทำการปรับปรุงคำตอบปัจจุบัน

ขั้นตอนที่ 9 หาคำตอบใหม่ที่ดีที่สุด และตรวจสอบว่าคำตอบใหม่ที่ดีที่สุดมีค่าน้อยกว่าคำตอบที่ดีที่สุดหรือไม่ ถ้าไม่น้อยกว่าจะให้จำนวนรอบการวนซ้ำเท่ากับจำนวนรอบการวนซ้ำบวก 1 แต่ถ้าน้อยกว่าจะให้ทำในขั้นตอนที่ 10

ขั้นตอนที่ 10 ให้ค่าคำตอบใหม่ที่ดีที่สุดไปแทนที่ค่าคำตอบปัจจุบัน

ขั้นตอนที่ 11 ให้จำนวนรอบปัจจุบันเท่ากับจำนวนรอบปัจจุบันบวก 1

ขั้นตอนที่ 12 ตรวจสอบว่าคำตอบใหม่ที่ดีที่สุดนั้นน้อยกว่าคำตอบที่ดีที่สุดหรือไม่ ถ้าไม่น้อยกว่าให้ จำนวนรอบปัจจุบันเท่ากับจำนวนรอบปัจจุบันบวก 1 แต่ถ้าน้อยกว่าให้ทำในขั้นตอนที่ 12 แต่ถ้าค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุด ให้ทำขั้นตอนที่ 13

ขั้นตอนที่ 13 ถ้าคำตอบใหม่ที่ดีที่สุดมีค่าน้อยกว่าคำตอบที่ดีที่สุด ให้ค่าคำตอบใหม่ที่ดีที่สุดไปแทนที่ค่าคำตอบที่ดีที่สุด

ขั้นตอนที่ 14 ถ้าจำนวนรอบปัจจุบันเท่ากับจำนวนรอบที่ผู้ใช้กำหนดแล้ว โปรแกรมจะหยุดการทำงาน

4.2 ความแตกต่างของ Local Search Type 1, Local Search Type 2 และ Iterated Local Search

4.2.1 Local Search Type 1 จะแตกต่างกับ Local Search Type 2 โดย Local Search Type 1 จะนำค่าคำตอบใหม่ที่ดีที่สุดในแต่ละรอบมาแทนที่คำตอบปัจจุบัน โดยไม่สนใจว่าคำตอบใหม่ที่จะไปแทนที่นั้นจะมีค่ามากกว่าหรือน้อยกว่าคำตอบปัจจุบันหรือไม่ แต่ถ้าคำตอบปัจจุบันนั้นมีค่าดีกว่าคำตอบที่ดีที่สุดก็จะนำคำตอบปัจจุบันไปแทนที่ในคำตอบที่ดีที่สุด แต่ถ้า Local Search Type 2 คำตอบใหม่ที่จะไปแทนที่คำตอบปัจจุบันได้นั้นจะต้องมีค่าคำตอบใหม่ที่น้อยกว่าค่าคำตอบปัจจุบัน ซึ่งคำตอบที่ดีที่สุดก็จะเท่ากับคำตอบปัจจุบัน

4.2.2 Local Search Type 2 จะแตกต่างกับ Iterated Local Search ตรงที่ Local Search Type 2 คำตอบใหม่ที่จะไปแทนที่คำตอบปัจจุบันได้นั้นจะต้องมีค่าคำตอบใหม่ที่น้อยกว่าค่าคำตอบปัจจุบัน ซึ่งจะทำให้ค่าคำตอบที่ดีที่สุดนั้นเท่ากับค่าคำตอบปัจจุบันด้วย แต่สำหรับ Iterated Local Search จะมีการกำหนดรอบการวนซ้ำของคำตอบปัจจุบัน ในกรณีที่ไม่พบคำตอบค่าคำตอบใหม่ที่ดีกว่าค่าคำตอบปัจจุบันแล้วจะให้วนซ้ำคำตอบปัจจุบันกี่รอบจึงจะทำการสุ่มคำตอบปัจจุบันขึ้นมาใหม่

แต่ถ้าคำตอบปัจจุบันมีค่าที่ดีกว่าคำตอบที่ดีที่สุดก็จะนำคำตอบปัจจุบันไปแทนที่ในคำตอบที่ดีที่สุดเช่นเดียวกัน

4.3 การใช้งานของโปรแกรม การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

4.3.1 เริ่มต้นเข้าสู่โปรแกรม

เมื่อเข้าสู่โปรแกรม จะพบเมนูหลัก ดังรูปที่ 4.12 ซึ่งแสดงหน้าแรกของโปรแกรม

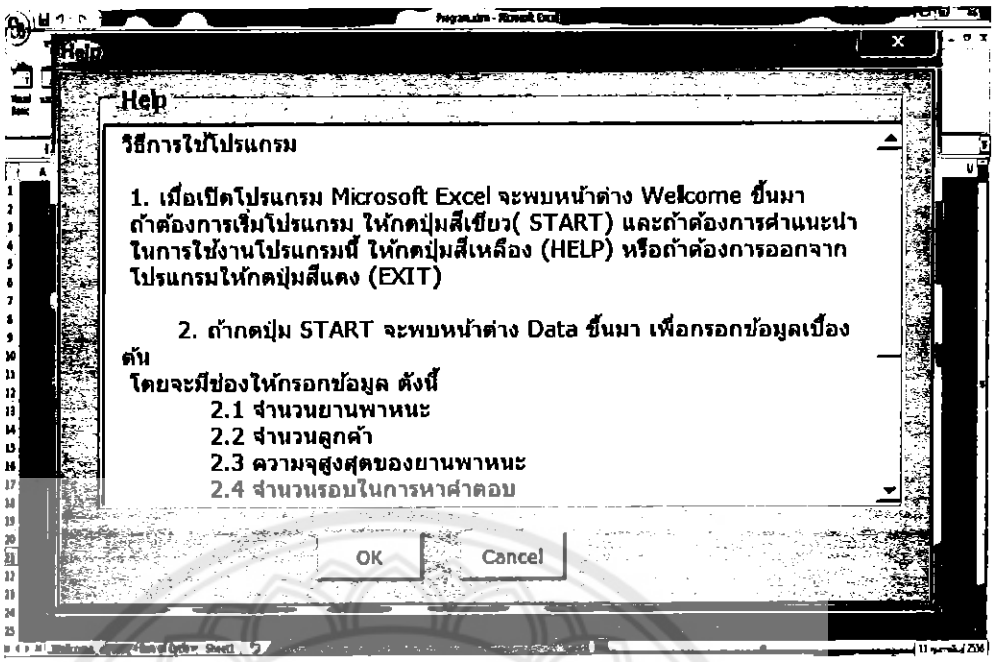
รูปที่ 4.12 หน้าแรกของโปรแกรม

ปุ่มของเมนูหลักจะมี 3 ปุ่ม ดังนี้

หมายเลข 1 ปุ่ม START เมื่อคลิ้กปุ่มนี้แล้วจะเข้าสู่ฟอร์ม Data เพื่อรับค่าที่ใช้ในการคำนวณ โดยการให้ผู้กรอกค่าต่างๆ ลงไป ดังรูปที่ 4.12

หมายเลข 2 ปุ่ม HELP เมื่อผู้ใช้คลิ้กปุ่มนี้ จะเข้าสู่หน้า User form Help ซึ่งมีหน้าที่ช่วยแนะนำวิธีการใช้โปรแกรม ดังแสดงในรูปที่ 4.12

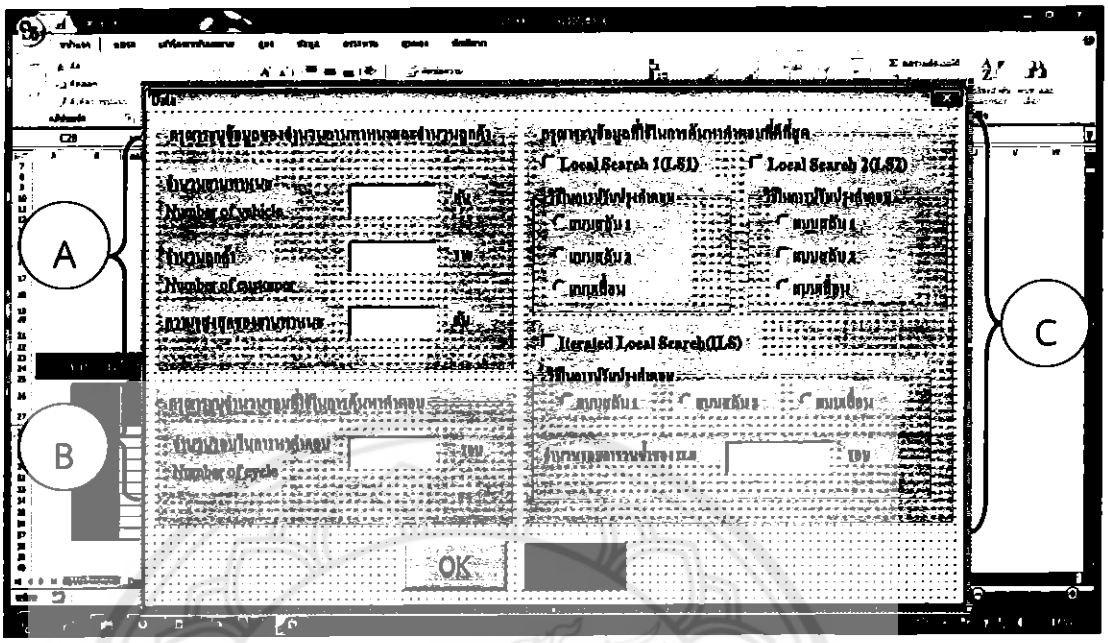
หมายเลข 3 ปุ่ม EXIT เมื่อผู้ใช้คลิ้กปุ่มนี้ จะเข้าสู่หน้า User form Confirm Exit เพื่อสอบถามผู้ใช้งานว่าต้องการออกจากโปรแกรมหรือไม่ ถ้าผู้ใช้กดปุ่ม OK จะคำสั่งให้ออกจากโปรแกรม แต่ถ้าผู้ใช้กดปุ่ม Cancel ก็จะกลับมายังหน้าหน้าแรกของโปรแกรม ดังแสดงในรูป 4.12



รูปที่ 4.13 User form Help ช่วยแนะนำวิธีการใช้โปรแกรม

รูปที่ 4.14 User form Confirm Exit เพื่อสอบถามผู้ใช้งานว่าต้องการออกจากโปรแกรมหรือไม่

4.3.2 ทำการกรอกข้อมูลต่างๆลงในแบบฟอร์ม Data



รูปที่ 4.15 การรับข้อมูลต่างๆ ของโปรแกรม

ในการกรอกข้อมูลในฟอร์ม Data นั้นแบ่งออกเป็น 4 ส่วน คือ

4.3.2.1 โซน A ให้ผู้ใช้กรอกข้อมูลจำนวนยานพาหนะ จำนวนลูกค้า และความจุสูงสุดของยานพาหนะ ลงในช่องกรอกข้อมูล

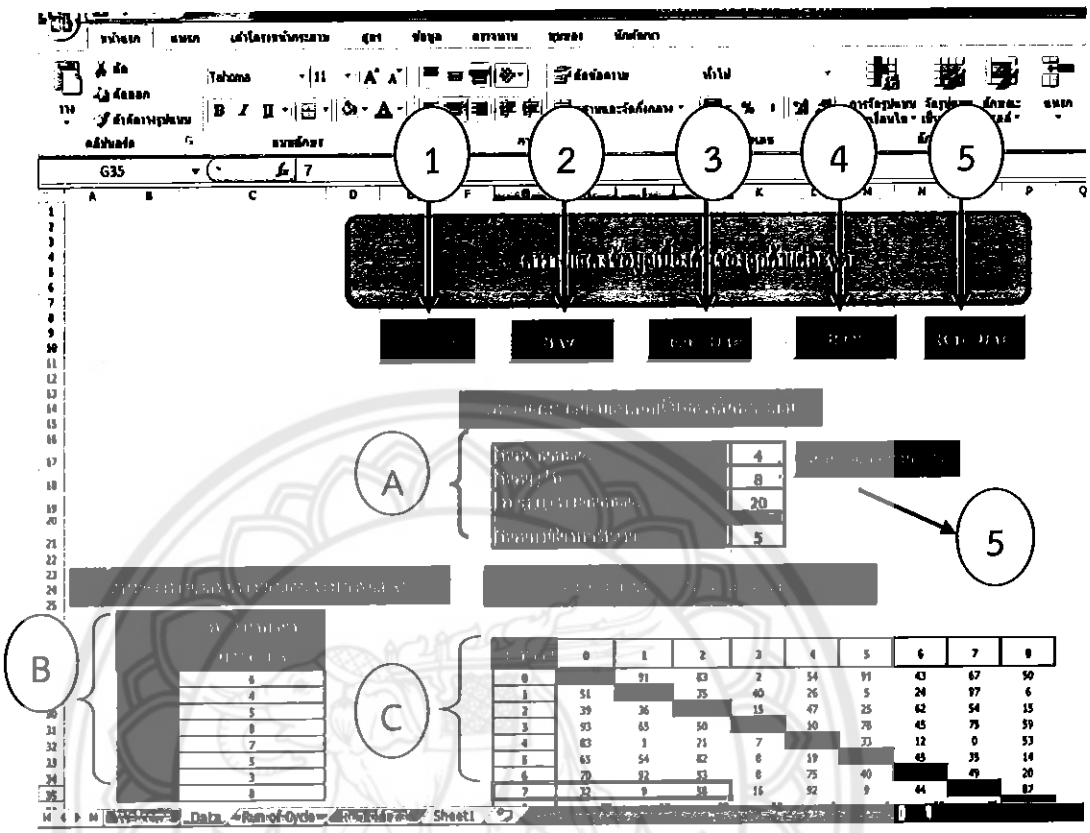
4.3.2.2 โซน B กรอกจำนวนรอบที่ใช้ในการค้นหาคำตอบ

4.3.2.3 โซน C เลือกวิธีที่ใช้ในการค้นหาคำตอบที่ดีที่สุด ซึ่งผู้ใช้สามารถเลือกวิธีในการในการค้นหาคำตอบได้ 1 วิธี จาก 3 วิธี อันได้แก่ Local Search 1 Local Search 2 และ Iterated Local Search จากนั้นทำการเลือกวิธีที่ใช้ในการปรับปรุงคำตอบซึ่งมีทั้งหมด 3 วิธี ได้แก่ แบบสลับ 1 แบบสลับ 2 และแบบเลื่อน ดังในรูปที่ 4.15 โซน C

4.3.2.3 หากผู้ใช้เลือกวิธีการค้นหาคำตอบแบบ Iterated Local Search จะต้องระบุจำนวนรอบในการวนซ้ำด้วย ดังรูปที่ 4.15 โซน C

เมื่อกรอกข้อมูลเสร็จสมบรูณ์ให้คลิ้ก OK ซึ่งข้อมูลที่กรอกนั้นจะแสดงผลที่ Work Sheet Data และ Work Sheet Run of Cycle เพื่อเข้าสู่ขั้นตอนต่อไป แต่ถ้าหากผู้ใช้ต้องการยกเลิกการกรอกข้อมูลใน User form Data นี้ ให้คลิ้กปุ่ม Cancel ได้ทันที

4.3.3 ตารางแสดงข้อมูลเบื้องต้นที่ใช้ในการค้นหาคำตอบที่ดีที่สุด



รูปที่ 4.16 แสดงผลข้อมูลที่ผู้ใช้กรอก

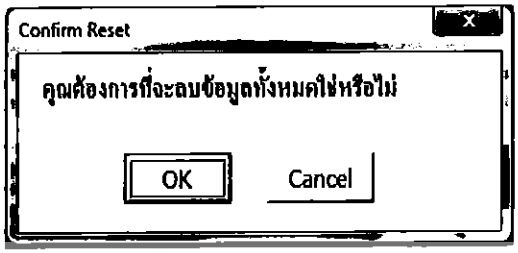
เมื่อทำการกรอกข้อมูลจาก User form Data เสร็จแล้วนั้น ถ้าผู้ใช้ต้องการเปลี่ยนแปลงค่าใหม่ ได้แก่ จำนวนยานพาหนะ จำนวนลูกค้า ความจุสูงสุดของยานพาหนะ และจำนวนรอบในการค้นหาคำตอบ ผู้ใช้สามารถแก้ไขค่าต่างๆ ในโซน A ได้ทันที แล้วกดปุ่มหมายเลข 5 เพื่อยืนยันข้อมูลเบื้องต้นที่แก้ไขแล้ว

4.4.3.1 โซน B เป็นความต้องการสินค้าของลูกค้าแต่ละจุดที่ผู้ใช้ต้องทำการกรอกลงไป

4.4.3.2 หมายเลข 1 ปุ่ม Check Data เป็นการตรวจสอบข้อมูลที่ผู้ใช้กรอกว่าถูกต้องหรือไม่ ซึ่งข้อมูลที่แสดงใน Work Sheet Data จะต้องเป็นตัวเลขเท่านั้น ดังรูปที่ 4.16 ถ้าผู้ใช้กรอกข้อมูลไม่ครบถ้วนหรือกรอกข้อมูลเป็นตัวหนังสือจะมีหน้าต่างแจ้งเตือนเพื่อให้ผู้ใช้แก้ไขให้เป็นตัวเลข และจะระบุว่าผู้ใช้จะต้องแก้ไขข้อมูลในโซนใด แต่สำหรับตารางแสดงข้อมูลเบื้องต้นนั้น จะต้องกรอกเป็นตัวเลขจำนวนเต็มเท่านั้น ซึ่งถ้าผู้ใช้กรอกข้อมูลเป็นตัวเลขทศนิยมก็จะมีหน้าต่างแจ้งเตือนเพื่อให้ผู้ใช้ได้แก้ไขข้อมูลในโซน B ดังรูปที่ 4.16

4.4.3.3 หมายเลข 2 ปุ่ม Save เมื่อผู้ใช้ต้องการที่จะบันทึกข้อมูลที่ได้กรอกไว้ สามารถคลิ้กปุ่มนี้ได้ทันที

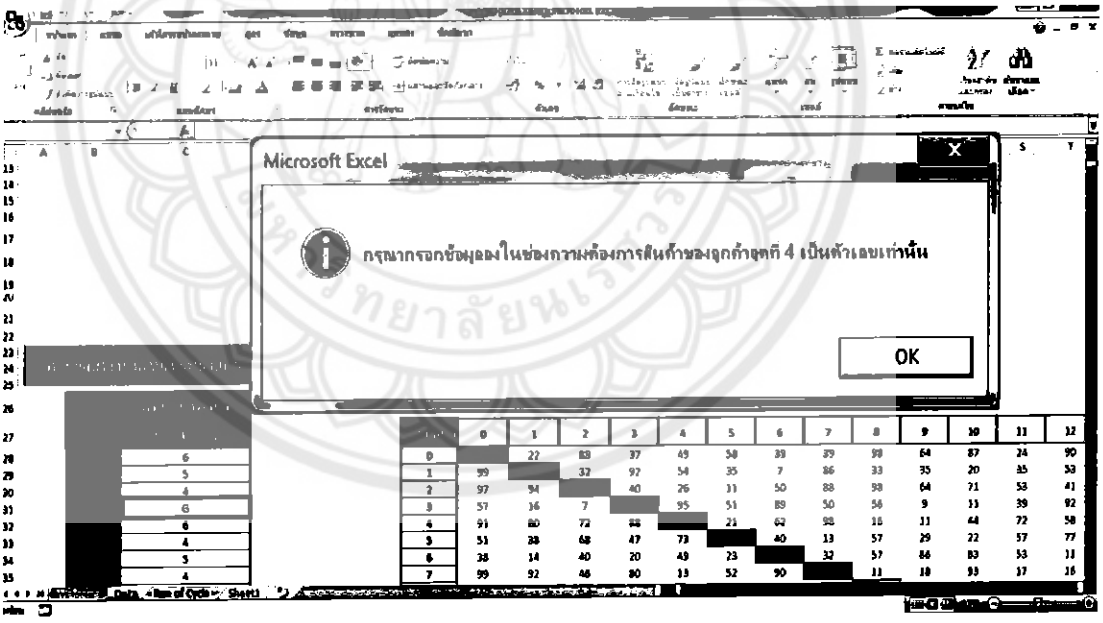
4.4.3.4 หมายเลข 3 ปุ่ม Clear Data ใช้ในการลบข้อมูลทั้งหมดของ Work Sheet Data ซึ่งเมื่อคลิ้กปุ่ม Clear Data จะมีหน้าต่างแสดงขึ้นมาเพื่อให้ยืนยันการลบข้อมูลทั้งหมด ดังรูปที่ 4.16 ถ้ากดปุ่ม OK ข้อมูลทั้งหมดก็จะถูกลบ แต่ถ้ากดปุ่ม Cancel ก็จะไปยัง Work Sheet Data



รูปที่ 4.17 หน้าต่างให้ยืนยันการลบข้อมูล ของปุ่ม Clear Data

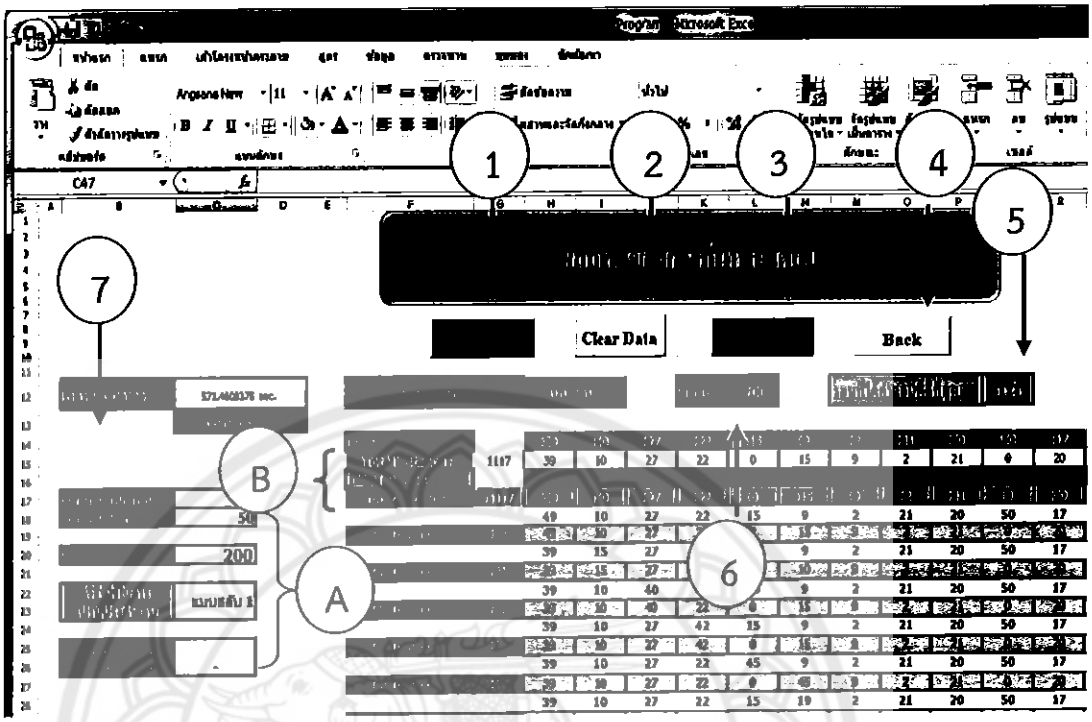
4.4.3.5 หมายเลข 4 ปุ่ม Back เมื่อคลิ้กปุ่มนี้แล้ว User form Data ก็แสดงขึ้นมา

4.4.3.6 หมายเลข 5 ปุ่ม Run Data หลังจากที่กรอกข้อมูลเรียบร้อยให้คลิ้กปุ่ม Run Data นี้ เพื่อทำการ Run โปรแกรมใน Work Sheet Run of Cycle



รูปที่ 4.18 หน้าต่างแจ้งเตือนเมื่อผู้ใช้กรอกข้อมูลเป็นตัวหนังสือ

4.3.4 ตารางแสดงผลลัพธ์ของการค้นหาคำตอบ



รูปที่ 4.19 แสดงผลลัพธ์ของการค้นหาคำตอบ

จากรูปที่ 4.19 จะแสดงผลลัพธ์ของการค้นหาคำตอบ มีรายละเอียดดังนี้

4.3.4.1 หมายเลข 1 ปุ่ม Run Data เป็นปุ่มที่จะสั่งให้โปรแกรมทำการ Run ข้อมูลทั้งหมดที่ผู้ใช้ได้กรอกข้อมูลไว้ในข้างต้น

4.3.4.2 หมายเลข 2 ปุ่ม Clear Data เป็นปุ่มที่ใช้ในการลบผลลัพธ์จากการ Run ทั้งหมด

4.3.4.3 หมายเลข 3 ปุ่ม Save เมื่อผู้ใช้ต้องการบันทึกผลลัพธ์จากการค้นหาคำตอบของโปรแกรมสามารถคลิ้กปุ่มนี้ได้ทันที

4.3.4.3 หมายเลข 4 ปุ่ม Back จะใช้เมื่อผู้ใช้ต้องการกลับไปยัง Work Sheet Data

4.3.4.4 หมายเลข 5 แสดงรอบที่ได้คำตอบที่ดีที่สุดจากจำนวนรอบในการค้นหาคำตอบทั้งหมด ว่าคำตอบที่ดีที่สุดนั้นถูกพบในการปรับปรุงคำตอบรอบที่เท่าไร

4.3.4.5 หมายเลข 6 แสดงจำนวนรอบในการค้นหาทั้งหมด

4.3.4.6 หมายเลข 7 แสดงเวลาทั้งหมดที่โปรแกรมใช้ในการค้นหาคำตอบที่ดีที่สุด ซึ่งมีทั้งหน่วยที่เป็นวินาที และหน่วยที่เป็นนาที

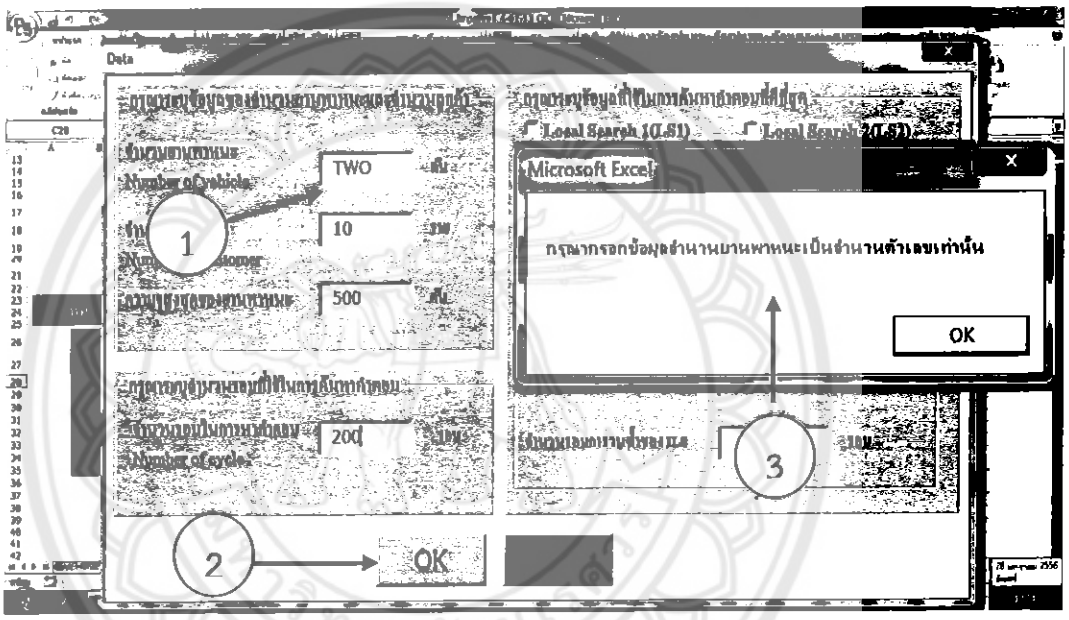
4.3.4.7 โซน A แสดงจำนวนยานพาหนะ จำนวนลูกค้า จำนวนรอบในการค้นหาคำตอบ วิธีที่ใช้ในการปรับปรุงคำตอบ และจำนวนรอบการวนซ้ำ ที่ผู้ใช้กรอกจาก User form Data

4.3.4.8 โซน B แสดงคำตอบที่ดีที่สุดที่ได้จากการค้นหาคำตอบ

4.3.5 ลักษณะการแจ้งข้อผิดพลาดที่เกิดขึ้นในการใช้โปรแกรม

ในการใช้โปรแกรม เมื่อมีข้อผิดพลาดเกิดขึ้นโปรแกรมจะมีการแจ้งเตือนให้ผู้ใช้ทราบเพื่อให้ผู้ใช้โปรแกรมได้ทำการแก้ไขข้อผิดพลาดที่เกิดขึ้นได้ ซึ่งลักษณะของการแจ้งเตือนข้อผิดพลาดจะมีลักษณะเป็นข้อความ ได้แก่ การแจ้งความผิดพลาดที่เกิดจากกรอกข้อมูลเป็นตัวหนังสือไม่ใช่ตัวเลข การแจ้งความผิดพลาดที่เกิดจากการกรอกค่าเป็นจุดทศนิยม และการแจ้งความผิดพลาดที่ผู้ใช้ลืมกรอกค่า

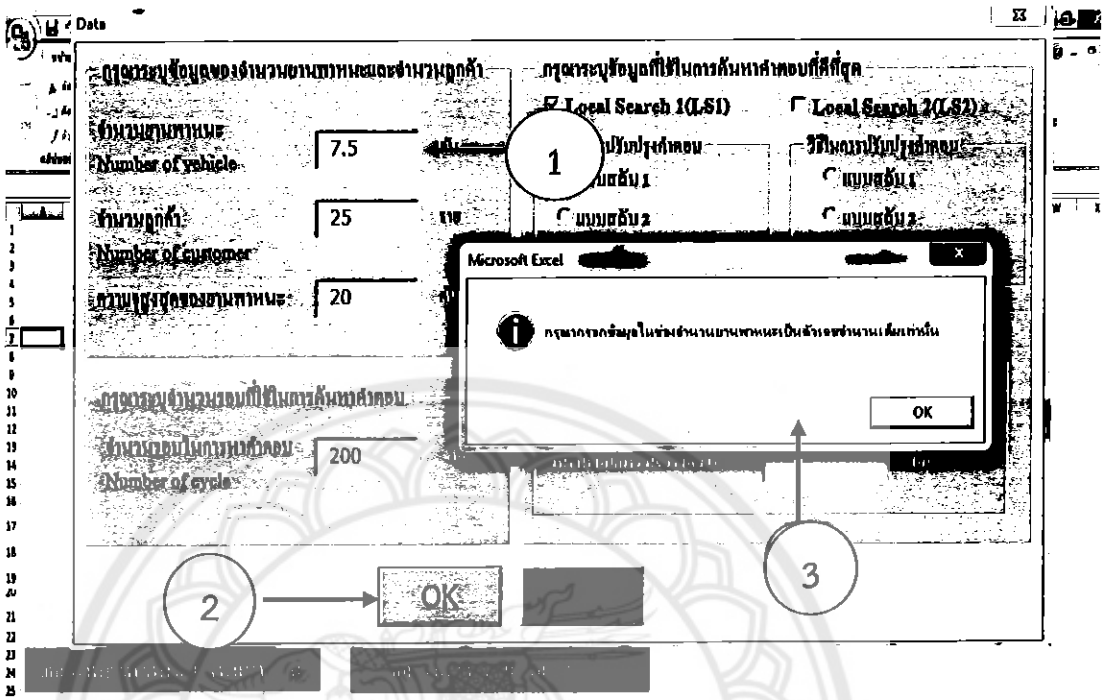
4.3.5.1 การแจ้งความผิดพลาดที่เกิดจากกรอกข้อมูลเป็นตัวหนังสือที่ไม่ใช่ตัวเลข จะมีหน้าต่าง จะแสดงการแจ้งเตือนให้กับผู้ใช้ว่าเกิดความผิดพลาดในการกรอกข้อมูลในลักษณะใด



รูปที่ 4.20 การแจ้งความผิดพลาดที่เกิดจากกรอกข้อมูลเป็นตัวหนังสือไม่ใช่ตัวเลข

จากรูปที่ 4.20 การแจ้งข้อผิดพลาดจะเกิดขึ้นเมื่อผู้ใช้ทำการกรอกข้อมูลในหมายเลข 1 เป็นตัวหนังสือ แล้วคลิกที่หมายเลข 2 ปุ่ม OK จะมีหน้าต่างแจ้งเตือนหมายเลข 3 ปรากฏขึ้น เพื่อเตือนให้ผู้ใช้ทราบข้อผิดพลาด จากนั้นให้ผู้ใช้คลิ้กที่ปุ่ม OK ในกรอบแจ้งเตือน เพื่อทำการแก้ไขข้อมูล

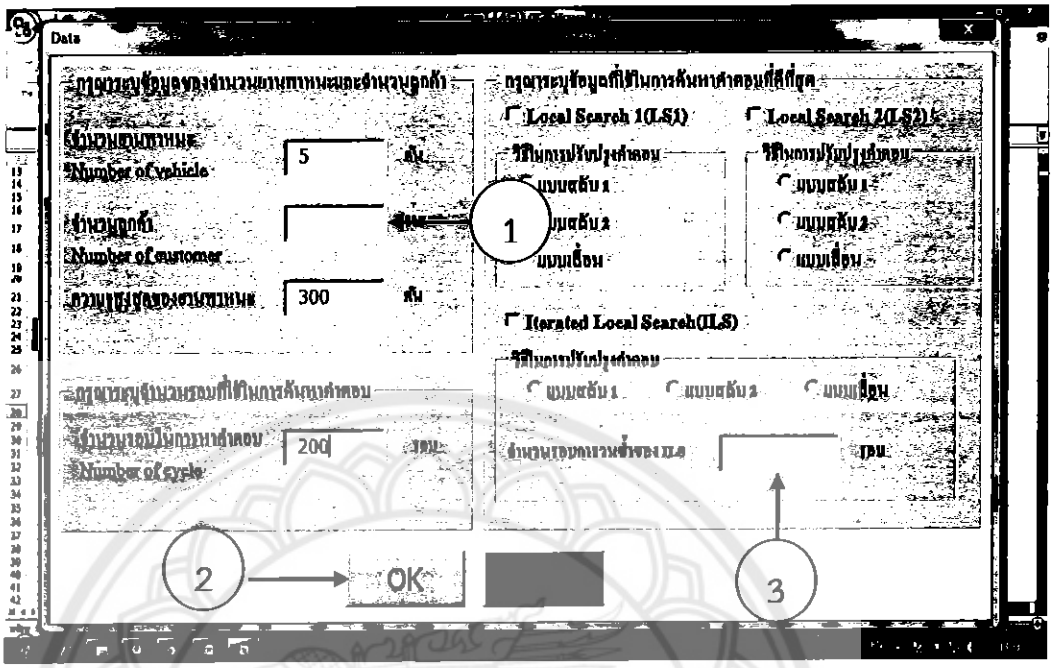
4.3.5.2 การแจ้งความผิดพลาดที่เกิดจากการกรอกค่าเป็นจุดทศนิยม



รูปที่ 4.21 การแจ้งความผิดพลาดที่เกิดจากการกรอกค่าเป็นจุดทศนิยม

จากรูปที่ 4.21 การแจ้งข้อผิดพลาดจะเกิดขึ้นเมื่อผู้ใช้ทำการกรอกข้อมูลในหมายเลข 1 เป็นจุดทศนิยม เมื่อคลิ้กที่หมายเลข 2 ปุ่ม OK จะมีหน้าต่างแจ้งเตือนหมายเลข 3 ปรากฏขึ้นเพื่อเตือนให้ผู้ใช้ทราบ จากนั้นให้ผู้ใช้คลิ้กที่ปุ่ม OK ในกรอบแจ้งเตือน เพื่อทำการแก้ไขข้อมูล

4.3.5.3 การแจ้งความผิดพลาดที่ผู้ใช้ลืมกรอกค่า



รูปที่ 4.22 การแจ้งความผิดพลาดที่ผู้ใช้ลืมกรอกค่า

จากรูปที่ 4.22 การแจ้งข้อผิดพลาดจะเกิดขึ้นเมื่อผู้ใช้ทำการคลิ้กหมายเลข 2 ปุ่ม OK ในขณะที่ผู้ใช้ลืมกรอกข้อมูลในช่องหมายเลข 1 โดยหน้าต่างหมายเลข 3 จะปรากฏขึ้น เพื่อเตือนให้ผู้ใช้ทราบ จากนั้นให้ผู้ใช้คลิ้กที่ปุ่ม OK ในหน้าต่างแจ้งเตือน เพื่อให้ทำการกรอกข้อมูลต่างๆ ให้ครบถ้วน

4.4 ผลการทดสอบโปรแกรม และวิเคราะห์ผลการทดลองในปัญหาขนาดเล็ก

หลังจากทำการทดสอบโปรแกรมแล้ว ต่อไปจะเป็นการแสดงผลลัพธ์ที่ได้จากการทดสอบประสิทธิภาพของโปรแกรมตามลักษณะของปัญหาขนาดเล็ก ทั้งหมด 2 ปัญหา

โดยคอมพิวเตอร์ที่ใช้ในการประมวลผลโปรแกรมการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก คือ Windows 7 Professional Intel Core i5 – 2430M CPU@2.40 GHz 2.40 GHz

การแสดงผลลัพธ์เมื่อมีการกำหนดค่าพารามิเตอร์ในการค้นหาคำตอบเฉพาะที่ของปัญหาขนาดเล็กเป็นการแสดงผลลัพธ์ของโปรแกรมกับค่าพารามิเตอร์ต่างๆ 2 ปัญหา โดยค่าพารามิเตอร์ คือ วิธีการปรับปรุงคำตอบ ได้แก่ การค้นหาคำตอบแบบสลับ 1 (Swap 1) การค้นหาคำตอบแบบสลับ 2 (Swap 2) และการค้นหาคำตอบแบบเลื่อน (Slide) ซึ่งผู้วิจัยได้ทำการเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆ ในการค้นหาคำตอบที่ดีที่สุดทั้ง 3 วิธี โดยให้มีจำนวนรอบในการหาคำตอบทั้งหมดเท่ากัน คือ 200 รอบ ซึ่งต่อไปจะเป็นการแสดงผลลัพธ์ของโปรแกรม ดังต่อไปนี้

ตารางที่ 4.2 ผลลัพธ์ของปัญหาที่ 1 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดเล็ก

หลักการพิจารณา	ระยะทางรวม (กิโลเมตร)			เวลาที่ใช้ในการประมวลผล (นาที)			Best Iteration Average
	ค่าเฉลี่ย	น้อยที่สุด	S.D	ค่าเฉลี่ย	น้อยที่สุด	S.D	
ปัญหาขนาดเล็ก : โจทย์ปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1							
1. แบบสลับ 1	320.2000	313	2.6300	0.4090	0.4080	0.0002	115
2. แบบสลับ 2	333.2000	313	6.5200	0.4160	0.4140	0.0005	109
3. แบบเลื่อน	345.4000	331	4.3000	0.4180	0.4180	0.0002	66
ปัญหาขนาดเล็ก : โจทย์ปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search Type 2							
1. แบบสลับ 1	359.4000	327	12.1700	0.4180	0.4140	0.0030	21
2. แบบสลับ 2	348	317	12.5000	0.4220	0.4180	0.0030	34
3. แบบเลื่อน	382	338	16.3900	0.4180	0.4110	0.0020	23
ปัญหาขนาดเล็ก : โจทย์ปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search							
1. แบบสลับ 1	342.4000	313	13.7700	0.4160	0.4070	0.0050	117
2. แบบสลับ 2	328	313	6.8500	0.4240	0.4200	0.0020	68
3. แบบเลื่อน	367.2000	338	12.1600	0.4220	0.4190	0.0009	41

จากตารางที่ 4.2 เป็นการแสดงผลลัพธ์ของปัญหาที่ 1 สำหรับเป็นปัญหาขนาดเล็ก ซึ่งจะแสดงระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะ ทั้งระยะทางรวมเฉลี่ย ระยะทางรวมที่น้อยที่สุด และค่าของส่วนเบี่ยงเบนมาตรฐาน เวลาที่ใช้ในการประเมินผลโปรแกรม ทั้งค่าเฉลี่ย ค่าที่น้อยที่สุด และค่าของส่วนเบี่ยงมาตรฐาน รวมทั้งรอบที่เจอคำตอบที่ดีที่สุดโดยเฉลี่ย เมื่อมีการกำหนดวิธีการค้นหาคำตอบเฉพาะที่เป็น Local Search Type 1, Local Search Type 2 และ Iterated Local Search ซึ่งจะแสดงตามหลักการที่ใช้ในการปรับปรุงคำตอบ ทั้ง 3 หลักการ

เมื่อพิจารณา ค่าระยะทางรวมเฉลี่ยในการขนส่งสินค้าให้กับลูกค้าของรถบรรทุกนั้น จะพบว่าจากวิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 การปรับปรุงคำตอบแบบสลับ 1 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำที่สุดเท่ากับ 320.2 กิโลเมตร และทั้งนี้ยังให้ค่าส่วนเบี่ยงมาตรฐาน (S.D) ที่

น้อยกว่าคำตอบแบบอื่นๆ คือ 2.63 ซึ่งบ่งบอกว่าคำตอบที่ได้นั้นมีความแม่นยำของคำตอบมากกว่าแบบอื่น และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 115

แต่เมื่อพิจารณาปัญหาขนาดเล็ก ของปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search Type 2 แล้ว พบว่า แบบสลับ 2 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำกว่าแบบสลับ 1 และแบบเลื่อน คือ 348 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 34

และสำหรับการพิจารณาปัญหาของปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search พบว่า แบบสลับ 2 จะให้ค่าระยะทางรวมเฉลี่ยต่ำกว่าแบบสลับ 1 และเลื่อน คือ 328 กิโลเมตร และมีค่าส่วนเบี่ยงเบนมาตรฐานที่ต่ำกว่าทั้งสองแบบ คือ 6.85 และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 68

จะเห็นได้ว่า ในปัญหาขนาดเล็ก โจทย์ปัญหาที่ 1 วิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 โดยวิธีการปรับปรุงคำตอบแบบสลับ 1 นั้น ให้คำตอบระยะทางเฉลี่ยที่ดีที่สุด ซึ่งมีค่าดีกว่าคำตอบที่ดีที่สุด อันดับ 2 ก็คือ วิธีการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search โดยวิธีการปรับปรุงคำตอบแบบสลับ 2 เท่ากับ 328 – 320.2 = 7.8 กิโลเมตร หรือเท่ากับ ร้อยละ 2.45 และทั้งนี้ยังให้ค่าส่วนเบี่ยงมาตรฐาน (S.D) ที่น้อยกว่าด้วย เท่ากับ 6.85 - 2.63 = 4.22 ซึ่งบ่งบอกว่าคำตอบที่ได้นั้นมีความแม่นยำของคำตอบมากกว่า

ตารางที่ 4.3 ผลลัพธ์ของปัญหาที่ 2 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดเล็ก

หลักการพิจารณา	ระยะทางรวม (กิโลเมตร)			เวลาที่ใช้ในการประมวลผล (นาที)			Best Iteration Average
	ค่าเฉลี่ย	น้อยที่สุด	S.D	ค่าเฉลี่ย	น้อยที่สุด	S.D	
ปัญหาขนาดเล็ก : โจทย์ปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1							
1. แบบสลับ 1	240.2000	223	5.4000	0.4040	0.4000	0.0020	104
2. แบบสลับ 2	252.6000	227	8.9100	0.4080	0.4050	0.0008	141
3. แบบเลื่อน	240	223	8.6300	0.4120	0.4070	0.0020	107
ปัญหาขนาดเล็ก : โจทย์ปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2							
1. แบบสลับ 1	271.2000	250	10.3100	0.4110	0.4090	0.0009	12
2. แบบสลับ 2	255.4000	223	9.8600	0.4150	0.4140	0.0004	58
3. แบบเลื่อน	335.2000	265	31.7500	0.4380	0.4110	0.0099	18
ปัญหาขนาดเล็ก : โจทย์ปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search							
1. แบบสลับ 1	267.4000	227	22.1900	0.4300	0.4190	0.004	61
2. แบบสลับ 2	255.2000	242	9.0100	0.4280	0.4210	0.002	42
3. แบบเลื่อน	270.8000	254	9.3900	0.4240	0.4210	0.002	109

จากตารางที่ 4.3 เป็นการแสดงผลลัพธ์ของปัญหาที่ 2 สำหรับเป็นปัญหาขนาดเล็ก ซึ่งจะแสดงระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะ ทั้งระยะทางรวมเฉลี่ย ระยะทางรวมที่น้อยที่สุด และค่าของส่วนเบี่ยงเบนมาตรฐาน เวลาที่ใช้ในการประเมินผลโปรแกรม ทั้งค่าเฉลี่ย ค่าที่น้อยที่สุด และค่าของส่วนเบี่ยงมาตรฐาน รวมทั้งรอบที่เจอคำตอบที่ดีที่สุดโดยเฉลี่ย เมื่อมีการกำหนดวิธีการค้นหาคำตอบเฉพาะที่เป็น Local Search Type 1, Local Search Type 2 และ Iterated Local Search ซึ่งจะแสดงตามหลักการที่ใช้ในการทำ Neighborhood Search ทั้ง 3 หลักการ

เมื่อพิจารณา ค่าระยะทางรวมเฉลี่ยในการขนส่งสินค้าให้กับลูกค้าของรถบรรทุกนั้น จะพบว่าแบบเลื่อน จากวิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำที่สุด คือ 240 กิโลเมตร แต่ก็ยังมีค่าที่ใกล้เคียงกับแบบสลับ 1 มากนั่น คือ 240.2 กิโลเมตร

แต่เมื่อพิจารณาปัญหาขนาดเล็ก ของปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search Type 2 แล้วพบว่าแบบสลับ 2 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำกว่าแบบสลับ 1 และแบบเลื่อน คือ 255.4 กิโลเมตร และมีค่าส่วนเบี่ยงเบนมาตรฐานที่ต่ำกว่าทั้งสองแบบ คือ 9.86 และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 58

และสำหรับการพิจารณาปัญหาของปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search พบว่า แบบสลับ 2 จะให้ค่าระยะทางรวมเฉลี่ยต่ำกว่าแบบสลับ 1 และเลื่อน คือ 255.2 กิโลเมตร และมีค่าส่วนเบี่ยงเบนมาตรฐานที่ต่ำกว่าทั้งสองแบบ คือ 9.01 และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 42

จะเห็นได้ว่า สำหรับปัญหาขนาดเล็กของโจทย์ปัญหาที่ 2 วิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 โดยการปรับปรุงคำตอบแบบเลื่อน จะให้คำตอบที่ดีที่สุดที่เท่ากับ 240 กิโลเมตร ซึ่งมีค่าดีกว่าค่าคำตอบที่ดีที่สุด อันดับที่ 2 นั่นคือ วิธีการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search จากวิธีการปรับปรุงคำตอบแบบสลับ 2 เท่ากับ 255.2 – 240 = 15.2 กิโลเมตร คิดเป็นร้อยละ 6.333 ซึ่งหากพิจารณาส่วนเบี่ยงเบนมาตรฐานแล้ว ผลต่างจะเท่ากับ 9.01 – 8.63 = 0.38

4.5 ผลการทดสอบโปรแกรม และวิเคราะห์ผลการทดลองในปัญหาขนาดกลาง

หลังจากทำการทดสอบโปรแกรมแล้ว ต่อไปจะเป็นการแสดงผลลัพธ์ที่ได้จากการทดสอบประสิทธิภาพของโปรแกรมตามลักษณะของปัญหาขนาดกลาง ทั้งหมด 2 ปัญหา

โดยคอมพิวเตอร์ที่ใช้ในการประมวลผลโปรแกรมการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก คือ Windows 7 Professional Intel Core i5

การแสดงผลลัพธ์เมื่อมีการกำหนดค่าพารามิเตอร์ในการค้นหาคำตอบเฉพาะที่ของปัญหาขนาดกลางเป็นการแสดงผลลัพธ์ของโปรแกรมกับค่าพารามิเตอร์ต่างๆ 2 ปัญหา โดยค่าพารามิเตอร์ คือ วิธีการปรับปรุงคำตอบ อันได้แก่ การค้นหาคำตอบแบบสลับ 1 (Swap 1) การค้นหาคำตอบแบบสลับ 2 (Swap 2) และ การค้นหาคำตอบแบบเลื่อน (Slide) ซึ่งผู้วิจัยได้ทำการเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆ ในการค้นหาคำตอบที่ดีที่สุดทั้ง 3 วิธี โดยให้มีจำนวนรอบในการหาคำตอบทั้งหมดเท่ากัน คือ 200 รอบ ซึ่งต่อไปจะเป็นการแสดงผลลัพธ์ของโปรแกรม ดังต่อไปนี้

ตารางที่ 4.4 ผลลัพธ์ของปัญหาที่ 1 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดกลาง

หลักการพิจารณา	ระยะทางรวม (กิโลเมตร)			เวลาที่ใช้ในการประมวลผล (นาที)			Best Iteration Average
	ค่าเฉลี่ย	น้อยที่สุด	S.D	ค่าเฉลี่ย	น้อยที่สุด	S.D	
ปัญหาขนาดกลาง : โจทย์ปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1							
1. แบบสลับ 1	575.800	545	14.430	2.250	2.230	0.010	113
2. แบบสลับ 2	670.600	653	5.960	2.180	2.080	0.046	105
3. แบบเลื่อน	618	578	15.880	2.120	2.110	0.002	150
ปัญหาขนาดกลาง : โจทย์ปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2							
1. แบบสลับ 1	655.200	537	48.480	2.150	2.080	0.027	50
2. แบบสลับ 2	546.600	511	13.720	2.160	2.110	0.010	164
3. แบบเลื่อน	700	517	58.730	2.180	2.140	0.010	73
ปัญหาขนาดกลาง : โจทย์ปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search							
1. แบบสลับ 1	562	506	19.690	2.220	2.190	0.010	140
2. แบบสลับ 2	625.200	6055	12.310	2.220	2.210	0.008	90
3. แบบเลื่อน	665.400	645	16.490	2.260	2.230	0.020	97

จากตารางที่ 4.4 เป็นการแสดงผลลัพธ์ของปัญหาที่ 1 สำหรับเป็นปัญหาขนาดกลาง ซึ่งจะแสดงระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะ ทั้งระยะทางรวมเฉลี่ย ระยะทางรวมที่น้อยที่สุด และค่าของส่วนเบี่ยงเบนมาตรฐาน เวลาที่ใช้ในการประเมินผลโปรแกรม ทั้งค่าเฉลี่ย ค่าที่น้อยที่สุด และค่าของส่วนเบี่ยงมาตรฐาน รวมทั้งรอบที่เจอคำตอบที่ดีที่สุดโดยเฉลี่ย เมื่อมีการกำหนดวิธีการค้นหาคำตอบเฉพาะที่เป็น Local Search Type 1, Local Search Type 2 และ Iterated Local Search ซึ่งจะแสดงตามหลักการที่ใช้ในการทำ Neighborhood Search ทั้ง 3 หลักการ

เมื่อพิจารณา ค่าระยะทางรวมเฉลี่ยในการขนส่งสินค้าให้กับลูกค้าของรถบรรทุกนั้น จะพบว่าจากวิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 2 การปรับปรุงคำตอบแบบสลับ 2 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำที่สุด คือ 546.6 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 164

แต่เมื่อพิจารณาปัญหาขนาดกลาง ของปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 แล้ว พบว่า แบบสลับ 1 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำกว่าแบบสลับ 2 และแบบเลื่อน คือ 575.8 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 113

และสำหรับการพิจารณาปัญหาของปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search พบว่า แบบสลับ 1 จะให้ค่าระยะทางรวมเฉลี่ยต่ำกว่าแบบสลับ 2 และเลื่อน คือ 562 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 140

จะเห็นได้ว่า จากปัญหาขนาดกลาง สำหรับโจทย์ปัญหาที่ 1 นั้น วิธีการค้นหาคำตอบเฉพาะที่ของ Local Search 2 โดยวิธีการปรับปรุงคำตอบแบบสลับ 2 จะให้ค่าคำตอบที่ดีที่สุด ซึ่งดีกว่าค่าที่ดีที่สุด อันดับ 2 นั่นคือ วิธีการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search จากวิธีการปรับปรุงคำตอบแบบสลับ 1 เท่ากับ 562 – 546.6 = 15.4 กิโลเมตร เมื่อคิดเป็นเปอร์เซ็นต์แล้วเท่ากับ ร้อยละ 2.80 และผลต่างของส่วนเบี่ยงเบนมาตรฐาน เท่ากับ 19.69 – 13.72 = 5.97

ตารางที่ 4.5 ผลลัพธ์ของปัญหาที่ 2 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดกลาง

หลักการพิจารณา	ระยะทางรวม (กิโลเมตร)			เวลาที่ใช้ในการประมวลผล (นาที)			Best Iteration Average
	ค่าเฉลี่ย	น้อยที่สุด	S.D	ค่าเฉลี่ย	น้อยที่สุด	S.D	
ปัญหาขนาดกลาง : โจทย์ปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1							
1. แบบสลับ 1	602.400	557	19.540	2.060	2.040	0.004	100
2. แบบสลับ 2	731.800	697	11.840	2.090	2.081	0.001	142
3. แบบเลื่อน	683.600	652	15.590	2.130	2.110	0.003	116
ปัญหาขนาดกลาง : โจทย์ปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2							
1. แบบสลับ 1	690	616	37.280	2.130	2.110	0.009	54
2. แบบสลับ 2	650.400	590	25.580	2.130	2.110	0.006	128
3. แบบเลื่อน	757	696	18.470	2.170	2.150	0.010	86

ตารางที่ 4.5 (ต่อ) ผลลัพธ์ของปัญหาที่ 2 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดกลาง

หลักการพิจารณา	ระยะทางรวม (กิโลเมตร)			เวลาที่ใช้ในการประมวลผล (นาที)			Best Iteration Average
	ค่าเฉลี่ย	น้อยที่สุด	S.D	ค่าเฉลี่ย	น้อยที่สุด	S.D	
ปัญหาขนาดกลาง : โจทย์ปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search							
1. แบบสลับ 1	636.200	600	13.550	2.180	2.120	0.030	147
2. แบบสลับ 2	737.800	625	65.700	2.170	2.500	0.010	135
3. แบบเลื่อน	763.600	692	39.620	2.270	2.190	0.020	87

จากตารางที่ 4.5 เป็นการแสดงผลลัพธ์ของปัญหาที่ 2 สำหรับเป็นปัญหาขนาดกลาง ซึ่งจะแสดงระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะ ทั้งระยะทางรวมเฉลี่ย ระยะทางรวมที่น้อยที่สุด และค่าของส่วนเบี่ยงเบนมาตรฐาน เวลาที่ใช้ในการประเมินผลโปรแกรม ทั้งค่าเฉลี่ย ค่าที่น้อยที่สุด และค่าของส่วนเบี่ยงมาตรฐาน รวมทั้งรอบที่เจอคำตอบที่ดีที่สุดโดยเฉลี่ย เมื่อมีการกำหนดวิธีการค้นหาคำตอบเฉพาะที่เป็น Local Search Type 1, Local Search Type 2 และ Iterated Local Search ซึ่งจะแสดงตามหลักการที่ใช้ในการทำ Neighborhood Search ทั้ง 3 หลักการ

เมื่อพิจารณา ค่าระยะทางรวมเฉลี่ยในการขนส่งสินค้าให้กับลูกค้าของรถบรรทุกนั้น จะพบว่า แบบสลับ 1 จากวิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำที่สุด คือ 602.4 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 100

แต่เมื่อพิจารณาปัญหาขนาดกลาง ของปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search Type 2 แล้ว พบว่า แบบสลับ 2 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำกว่าแบบสลับ 1 และแบบเลื่อน คือ 650.4 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 128

และสำหรับการพิจารณาปัญหาของปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search พบว่า แบบสลับ 1 จะให้ค่าระยะทางรวมเฉลี่ยต่ำกว่าแบบสลับ 2 และเลื่อน คือ 636.2 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 147

จะเห็นได้ว่า ในปัญหาขนาดกลางของโจทย์ปัญหาที่ 2 วิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 ด้วยการปรับปรุงคำตอบแบบสลับ 1 จะให้ค่าคำตอบที่ดีที่สุด ซึ่งดีกว่าค่าคำตอบที่ดีที่สุด อันดับ 2 จากวิธีการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search โดยการปรับปรุงคำตอบแบบสลับ 1 เท่ากับ 636.2 – 602.4 = 33.8 กิโลเมตร ซึ่งสามารถคิดเป็นร้อยละได้เท่ากับ

ร้อยละ 5.61 และหากพิจารณาส่วนเบี่ยงเบนมาตรฐานแล้ววิธีการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search จะดีกว่าวิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 เท่ากับ 19.54 – 13.55 = 5.99

4.6 ผลการทดสอบโปรแกรม และวิเคราะห์ผลการทดลองในปัญหาขนาดใหญ่

หลังจากทำการทดสอบโปรแกรมแล้ว ต่อไปจะเป็นการแสดงผลลัพธ์ที่ได้จากการทดสอบประสิทธิภาพของโปรแกรมตามลักษณะของปัญหาขนาดใหญ่ ทั้งหมด 6 ปัญหา

โดยคอมพิวเตอร์ที่ใช้ในการประมวลผลโปรแกรมการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก คือ Windows 7 Professional Intel Core i5

การแสดงผลลัพธ์เมื่อมีการกำหนดค่าพารามิเตอร์ในการค้นหาคำตอบเฉพาะที่ของปัญหาขนาดใหญ่เป็นการแสดงผลลัพธ์ของโปรแกรมกับค่าพารามิเตอร์ต่างๆ 2 ปัญหา โดยค่าพารามิเตอร์ คือ วิธีการปรับปรุงคำตอบ อันได้แก่ การค้นหาคำตอบแบบสลับ 1 (Swap 1) การค้นหาคำตอบแบบสลับ 2 (Swap 2) และ การค้นหาคำตอบแบบเลื่อน (Slide) ซึ่งผู้วิจัยได้ทำการเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆ ในการค้นหาคำตอบที่ดีที่สุดทั้ง 3 วิธี โดยให้มีจำนวนรอบในการหาคำตอบทั้งหมดเท่ากัน คือ 200 รอบ ซึ่งต่อไปจะเป็นการแสดงผลลัพธ์ของโปรแกรม ดังต่อไปนี้

ตารางที่ 4.6 ผลลัพธ์ของปัญหาที่ 1 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดใหญ่

หลักการพิจารณา	ระยะทางรวม (กิโลเมตร)			เวลาที่ใช้ในการประมวลผล (นาที)			Best Iteration Average
	ค่าเฉลี่ย	น้อยที่สุด	S.D	ค่าเฉลี่ย	น้อยที่สุด	S.D	
ปัญหาขนาดใหญ่ : โจทย์ปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1							
1. แบบสลับ 1	1,106.400	1,068	23.370	7.690	7.620	0.022	152
2. แบบสลับ 2	1,370.600	1,301	29.070	7.710	7.680	0.010	142
3. แบบเลื่อน	1,352	1,249	47.990	7.880	7.860	0.010	133

ตารางที่ 4.6 (ต่อ) ผลลัพธ์ของปัญหาที่ 1 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดใหญ่

หลักการพิจารณา	ระยะทางรวม (กิโลเมตร)			เวลาที่ใช้ในการประมวลผล (นาที)			Best Iteration Average
	ค่าเฉลี่ย	น้อยที่สุด	S.D	ค่าเฉลี่ย	น้อยที่สุด	S.D	
ปัญหาขนาดใหญ่ : โจทย์ปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2							
1. แบบสลับ 1	1,141.800	1,073	27.750	7.460	7.410	0.015	127
2. แบบสลับ 2	1,230.600	1,115	40.880	7.510	7.480	0.010	166
3. แบบเลื่อน	1,417.400	1,311	36.520	7.790	7.720	0.020	185
ปัญหาขนาดใหญ่ : โจทย์ปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search							
1. แบบสลับ 1	1,220.600	1,183	18.020	7.560	7.300	0.080	77
2. แบบสลับ 2	1,355.200	1,255	36.470	7.670	7.180	0.140	51
3. แบบเลื่อน	1,577.600	1,492	34.020	7.300	7.230	0.028	109

จากตารางที่ 4.6 เป็นการแสดงผลลัพธ์ของปัญหาที่ 1 สำหรับเป็นปัญหาขนาดใหญ่ ซึ่งจะแสดงระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะ ทั้งระยะทางรวมเฉลี่ย ระยะทางรวมที่น้อยที่สุด และค่าของส่วนเบี่ยงเบนมาตรฐาน เวลาที่ใช้ในการประเมินผลโปรแกรม ทั้งค่าเฉลี่ย ค่าที่น้อยที่สุด และค่าของส่วนเบี่ยงมาตรฐาน รวมทั้งรอบที่เจอคำตอบที่ดีที่สุดโดยเฉลี่ย เมื่อมีการกำหนดวิธีการค้นหาคำตอบเฉพาะที่เป็น Local Search Type 1, Local Search Type 2 และ Iterated Local Search ซึ่งจะแสดงตามหลักการที่ใช้ในการทำ Neighborhood Search ทั้ง 3 หลักการ

เมื่อพิจารณา ค่าระยะทางรวมเฉลี่ยในการขนส่งสินค้าให้กับลูกค้าของรถบรรทุกนั้น จะพบว่า แบบสลับ 1 จากวิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำที่สุด คือ 1106.4 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 152

แต่เมื่อพิจารณาปัญหาขนาดกลาง ของปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search Type 2 แล้ว พบว่า แบบสลับ 1 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำกว่าแบบสลับ 2 และแบบเลื่อน คือ 1141.8 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 127

และสำหรับการพิจารณาปัญหาของปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search พบว่า แบบสลับ 1 จะให้ค่าระยะทางรวมเฉลี่ยต่ำกว่าแบบสลับ 2 และเลื่อน คือ 1220.6 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 77

จะเห็นได้ว่า สำหรับปัญหาขนาดใหญ่ของโจทย์ปัญหาที่ 1 นี้ วิธีการค้นหาคำตอบเฉพาะที่ของ Local Search 1 ด้วยวิธีการปรับปรุงคำตอบแบบสลับ 1 จะให้ค่าที่ดีที่สุดที่ดีกว่า วิธีการค้นหาคำตอบเฉพะที่ของ Local Search 2 สำหรับวิธีการปรับปรุงคำตอบแบบสลับ 1 เท่ากับ 1,141.8 – 1,106.4 = 35.4 กิโลเมตร ซึ่งคิดเป็นร้อยละ 3.20 และมีผลต่างของส่วนเบี่ยงเบนมาตรฐาน เท่ากับ 27.75 – 23.37 = 4.38

ตารางที่ 4.7 ผลลัพธ์ของปัญหาที่ 2 เมื่อมีรูปแบบการค้นหาคำตอบที่ดีที่สุดที่แตกต่างกัน 3 รูปแบบของปัญหาขนาดใหญ่

หลักการพิจารณา	ระยะทางรวม (กิโลเมตร)			เวลาที่ใช้ในการประมวลผล (นาที)			Best Iteration Average
	ค่าเฉลี่ย	น้อยที่สุด	S.D	ค่าเฉลี่ย	น้อยที่สุด	S.D	
ปัญหาขนาดใหญ่ : โจทย์ปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1							
1. แบบสลับ 1	1,232.400	1,191	25.240	7.790	7.680	0.040	123
2. แบบสลับ 2	1,472.600	1,454	7.940	7.820	7.660	0.060	130
3. แบบเลื่อน	1,520	1,379	51.610	7.880	7.850	0.009	145
ปัญหาขนาดใหญ่ : โจทย์ปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2							
1. แบบสลับ 1	1,228	1,117	41.730	7.760	7.520	0.020	138
2. แบบสลับ 2	1,287.200	1,129	47.620	7.570	7.530	0.010	177
3. แบบเลื่อน	1,580.400	1,412	80.090	7.780	7.750	0.008	113
ปัญหาขนาดใหญ่ : โจทย์ปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search							
1. แบบสลับ 1	1,288.600	1,233	28.770	7.300	7.010	0.140	56
2. แบบสลับ 2	1,438.400	1,383	28.110	7.750	7.680	0.027	83
3. แบบเลื่อน	1,594	1,490	41.210	7.360	7.110	0.178	147

จากตารางที่ 4.7 เป็นการแสดงผลลัพธ์ของปัญหาที่ 2 สำหรับเป็นปัญหาขนาดใหญ่ ซึ่งจะแสดงระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะ ทั้งระยะทางรวมเฉลี่ย ระยะทางรวมที่น้อยที่สุด และค่าของส่วนเบี่ยงเบนมาตรฐาน เวลาที่ใช้ในการประเมินผลโปรแกรม ทั้งค่าเฉลี่ย ค่าที่น้อยที่สุด และค่าของส่วนเบี่ยงมาตรฐาน รวมทั้งรอบที่เจอคำตอบที่ดีที่สุดโดยเฉลี่ย เมื่อมีการกำหนดวิธีการค้นหาคำตอบเฉพาะที่เป็น Local Search Type 1, Local Search Type 2 และ Iterated Local Search ซึ่งจะแสดงตามหลักการที่ใช้ในการทำ Neighborhood Search ทั้ง 3 หลักการ

เมื่อพิจารณา ค่าระยะทางรวมเฉลี่ยในการขนส่งสินค้าให้กับลูกค้าของรถบรรทุกนั้น จะพบว่าแบบสลับ 1 จากวิธีการค้นหาคำตอบเฉพาะที่ของ Local Search Type 2 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำที่สุดของการเปลี่ยนแปลงค่าพารามิเตอร์ทั้ง 3 แบบ คือ 1,228 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 138

แต่เมื่อพิจารณาปัญหาขนาดใหญ่ ของปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search Type 1 แล้ว พบว่า แบบสลับ 1 จะให้ค่าระยะทางรวมเฉลี่ยที่ต่ำกว่าแบบสลับ 2 และแบบเลื่อน คือ 1,232.4กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 123

และสำหรับการพิจารณาปัญหาของปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search พบว่า แบบสลับ 1 จะให้ค่าระยะทางรวมเฉลี่ยต่ำกว่าแบบสลับ 2 และเลื่อน คือ 1,288.6 กิโลเมตร และพบคำตอบที่ดีที่สุดเฉลี่ยรอบที่ 56

จะเห็นได้ว่า ในปัญหาขนาดใหญ่ของโจทย์ปัญหาที่ 1 จากวิธีการค้นหาคำตอบเฉพาะที่ของ Local Search 2 โดยวิธีการปรับปรุงคำตอบแบบสลับ 1 จะให้ค่าคำตอบที่ดีกว่าค่าคำตอบที่ดีที่สุดอันดับที่ 2 เท่ากับ 1,232.4 – 1,228 = 4.4 กิโลเมตร ซึ่งคิดเป็นร้อยละได้เท่ากับ ร้อยละ 0.36

4.7 การวิเคราะห์ผลการทดลอง

ในการวิเคราะห์ผลการทดลองนี้ จะทำการวิเคราะห์ และเปรียบเทียบผลการทดลองที่ได้จากการประมวลผลจากโปรแกรมการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก และจากการเปรียบเทียบผลนั้น สามารถวิเคราะห์ และเปรียบเทียบผลการทดลองได้เป็น 2 ประเด็น คือ การเปรียบเทียบผลจากหลักการในการปรับปรุงคำตอบทั้ง 3 หลักการกับระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะ, ขนาดของปัญหากับระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะและเวลาในการประมวลผล และการวิเคราะห์ความแปรปรวน (ANOVA) ของผลลัพธ์โดยใช้โปรแกรม Minitab 16

4.7.1 การเปรียบเทียบผลจากหลักการในการปรับปรุงคำตอบทั้ง 3 หลักการกับระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะ

จากหลักการในการปรับปรุงคำตอบของทั้ง 3 หลักการ อันได้แก่ หลักการแบบสลับ 1 หลักการแบบสลับ 2 และหลักการแบบเลื่อน เมื่อทำการประมวลโปรแกรมด้วยปัญหาต่างๆ แล้วพบว่า

หลักการแบบสลับ 1 ให้คำตอบระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะเฉลี่ย และระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะที่น้อยที่สุด ดีกว่าหลักการแบบสลับ 2 และหลักการแบบเลื่อน ในปัญหาขนาดเล็กของปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 ซึ่งสามารถดูค่าระยะทางได้จากตารางที่ 4.1 ในปัญหาขนาดกลางของปัญหาที่ 1 และปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 และการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search ซึ่งสามารถดูค่าระยะทางได้จากตารางที่ 4.4 และตารางที่ 4.5 และในปัญหาขนาดใหญ่ของปัญหาที่ 1 และปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 การค้นหาคำตอบเฉพาะที่ของ Local Search 2 และการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search ซึ่งสามารถดูได้จากตารางที่ 4.6 และตารางที่ 4.7

หลักการแบบสลับ 2 ให้คำตอบระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะเฉลี่ย และระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะที่น้อยที่สุด ดีกว่าหลักการแบบสลับ 1 และหลักการแบบเลื่อน ในปัญหาขนาดเล็กของปัญหาที่ 1 และปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2 และการค้นหาคำตอบเฉพาะที่ของ Iterated Local Search ซึ่งสามารถดูได้จากตารางที่ 4.2 และตารางที่ 4.3 และในปัญหาขนาดกลางของปัญหาที่ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2 ซึ่งสามารถดูได้จากตารางที่ 4.4

หลักการแบบเลื่อน ให้คำตอบระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะเฉลี่ย และระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะที่น้อยที่สุด ดีกว่าหลักการแบบสลับ 1 และหลักการแบบสลับ 2 ในปัญหาขนาดเล็กของปัญหาที่ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 ซึ่งสามารถดูได้จากตารางที่ 4.3

ซึ่งสามารถสรุปผลได้ดังตารางที่ 4.8 ซึ่งเป็นตารางแสดงผลลัพธ์จากการคำตอบที่ดีที่สุดของแต่ละวิธีการที่ใช้ในการค้นหาคำตอบที่ดีที่สุด

ตารางที่ 4.8 แสดงผลลัพธ์จากการคำตอบที่ดีที่สุดของแต่ละวิธีการที่ใช้ในการค้นหาคำตอบที่ดีที่สุด

โจทย์	หลักการ	ปัญหาขนาดเล็ก			ปัญหาขนาดกลาง			ปัญหาขนาดใหญ่		
		LS1	LS2	ILS	LS1	LS2	ILS	LS1	LS2	ILS
ปัญหาที่ 1	Swap1									
	Swap2									
	Slide									
ปัญหาที่ 2	Swap1									
	Swap2									
	Slide									

4.7.2 ขนาดของปัญหากับระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะและเวลาในการประมวลผล

ปัญหาขนาดเล็ก จะเห็นได้ว่าค่าคำตอบซึ่งเป็นระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะมีค่าคำตอบที่ใกล้เคียงของวิธีการค้นหาคำตอบทั้ง 3 วิธี เนื่องจากปัญหาขนาดเล็กมีความหลากหลายน้อยกว่าปัญหาขนาดอื่นๆ จึงมีโอกาสที่จะเจอคำตอบในบริเวณเดียวกันได้สูง และใช้เวลาไม่มากสำหรับการค้นหาคำตอบที่ดีที่สุด เพราะปัญหาขนาดเล็กนั้น กำหนดให้มียานพาหนะจำนวน 3 คัน และลูกค้าจำนวน 10 ราย

ปัญหาขนาดกลาง ระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะของทั้ง 2 ปัญหานั้น เริ่มมีความหลากหลายมากขึ้น เนื่องจากกำหนดให้มียานพาหนะจำนวน 7 คัน และลูกค้าจำนวน 25 ราย ส่งผลให้ใช้เวลาในการประมวลผลที่มากขึ้นและระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะมากขึ้นตามไปด้วย

ปัญหาขนาดใหญ่ เป็นปัญหาที่มีระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะหลากหลายมากที่สุด ซึ่งได้กำหนดให้มียานพาหนะจำนวน 13 คัน และลูกค้าจำนวน 50 ราย จึงส่งผลมีการกระจายตัวของคำตอบ และใช้เวลาในการประมวลผลมากกว่าปัญหาขนาดอื่นๆ

4.7.3 การวิเคราะห์ความแปรปรวน (ANOVA) ของผลลัพธ์ในการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก โดยใช้โปรแกรม Minitab 16

การวิเคราะห์ความแปรปรวนเป็นหนึ่งในวิธีการทางสถิติที่นิยมใช้กันอย่างทั่วไปในการตัดสินใจทางสถิติคือการทดสอบสมมติฐาน (Hypothesis Testing) ซึ่งในโปรแกรม Minitab 16 นั้นจะมีคำสั่งในเรื่องการทดสอบสมมติฐานมากมาย รวมถึงการวิเคราะห์ความแปรปรวน โดยทั่วไปการ

ทดสอบสมมติฐานจะสันนิษฐานว่าสิ่งที่เราสนใจนั้นเป็นจริงก่อน แล้วจึงทำการพิสูจน์ด้วยข้อมูลจากกลุ่มตัวอย่าง

การทดสอบสมมติฐานประกอบด้วย 2 ส่วน คือ สมมติฐานหลัก และสมมติฐานรอง สำหรับสมมติฐานหลักนั้นจะเป็นข้อสันนิษฐานเบื้องต้น ซึ่งมักอ้างอิงจากผลการศึกษาก่อนหน้านี้หรือความรู้ทั่วไป ส่วนสมมติฐานรองคือสิ่งที่ต้องการพิสูจน์ว่าเป็นจริง

ซึ่งการวิเคราะห์ความแปรปรวนนี้ จะเป็นการวิเคราะห์ว่า มีความแตกต่างระหว่างผลลัพธ์ของค่าระยะทางในการเดินทางของยานพาหนะของการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก ในแต่ละวิธีการของการปรับปรุงคำตอบ อันได้แก่ แบบสลับ 1 แบบสลับ 2 และแบบเลื่อน ซึ่งเป็นปัจจัยที่มีผลกระทบของค่าของผลลัพธ์ อย่างมีนัยสำคัญทางสถิติหรือไม่ สามารถตรวจสอบได้โดยการวิเคราะห์ความแปรปรวนแบบปัจจัยเดียว (One – Way ANOVA) โดยเป็นการทดสอบว่าค่าเฉลี่ยสำหรับข้อมูลสองกลุ่มขึ้นไป ที่มีการแบ่งกลุ่มโดยปัจจัยเดียวนั้นเท่ากันหรือไม่

4.7.3.1 การแปรผลจากโปรแกรม Minitab 16

เป็นการแปรผลจากโปรแกรม Minitab 16 จากการวิเคราะห์ความแปรปรวนระหว่างค่าเฉลี่ยของระยะทางในการเดินทางของยานพาหนะในแต่ละวิธีการของการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก ซึ่งกระบวนการตัดสินใจสำหรับการทดสอบสมมติฐานนั้นอยู่บนพื้นฐานของความน่าจะเป็นทางสถิติหรือเราเรียกว่าค่า P – Value ที่ได้จากการทดสอบ

ก. ถ้าค่า P – Value มีค่าน้อยกว่าหรือเท่ากับระดับนัยสำคัญ (α - Level) ซึ่งเท่ากับ 0.05 ให้ปฏิเสธสมมติฐานหลัก ซึ่งก็คือ วิธีการปรับปรุงคำตอบไม่มีผลกระทบต่อการหาค่าคำตอบ และยอมรับสมมติฐานรอง ซึ่งก็คือ วิธีการมีผลต่อการหาค่าคำตอบ

ข. ถ้าค่า P – Value มีค่ามากกว่าระดับนัยสำคัญ (α - Level) ซึ่งเท่ากับ 0.05 แสดงว่าไม่สามารถปฏิเสธสมมติฐานหลักได้ จึงไม่สามารถยอมรับสมมติฐานรอง

4.7.3.2 สรุปผลการวิเคราะห์ความแปรปรวนระหว่างค่าเฉลี่ยของระยะทางในการเดินทางของยานพาหนะในแต่ละวิธีการ

ค่าความน่าจะเป็นทางสถิติของโจทย์ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุซึ่งมีขนาดปัญหา 3 ขนาด ได้แก่ ปัญหาขนาดเล็ก มีลูกค้า 10 ราย, ปัญหาขนาดกลาง มีลูกค้า 25 ราย และปัญหาขนาดใหญ่ มีลูกค้า 50 ราย แสดงไว้ดังตารางที่ 4.8 และกำหนดให้ระดับนัยสำคัญ มีค่าเท่ากับ 0.05

ตารางที่ 4.9 สรุปผลการวิเคราะห์ความแปรปรวน

รายละเอียด	Local Search 1	Local Search 2	Iterated Local Search
	P – Value	P – Value	P – Value
1. ปัญหาขนาดเล็ก โจทย์ปัญหาที่ 1	0.004	0.184	0.052
2. ปัญหาขนาดเล็ก โจทย์ปัญหาที่ 2	0.374	0.020	0.692
3. ปัญหาขนาดกลาง โจทย์ปัญหาที่ 1	0.000	0.050	0.001
4. ปัญหาขนาดกลาง โจทย์ปัญหาที่ 2	0.000	0.035	0.100
5. ปัญหาขนาดใหญ่ โจทย์ปัญหาที่ 1	0.000	0.000	0.000
6.ปัญหาขนาดใหญ่ โจทย์ปัญหาที่ 2	0.000	0.001	0.000

จากตารางที่ 4.9 แสดงให้เห็นว่าค่า P – Value ในปัญหาขนาดเล็ก ปัญหาที่ 1 ของ Local Search 2 และ Iterated Local Search ในปัญหาที่ 2 Local Search 1 และ Iterated Local Search มีค่ามากกว่านัยสำคัญที่เท่ากับ 0.05 แสดงว่าไม่สามารถปฏิเสธสมมติฐานหลักซึ่งก็คือ วิธีการหาคำตอบทั้งแบบสลับ 1 แบบสลับ 2 และแบบเลื่อนที่ใช้นั้น ไม่มีผลกระทบต่อการหาค่าคำตอบได้ จึงเป็นเหตุให้ไม่สามารถยอมรับสมมติฐานรอง ที่วิธีการหาค่าคำตอบ มีผลต่อการหาคำตอบตามไปด้วย ซึ่งในปัญหาขนาดเล็ก ปัญหาที่ 1 ของ Local Search 1 และปัญหาที่ 2 ของ Local Search 2 นั้น P – Value มีค่าน้อยกว่าระดับนัยสำคัญที่เท่ากับ 0.05 แสดงว่าสามารถปฏิเสธสมมติฐานหลัก และยอมรับสมมติฐานรอง ซึ่งก็คือ วิธีการปรับปรุงคำตอบทั้งแบบสลับ 1 แบบสลับ 2 และแบบเลื่อนที่ใช้นั้น มีผลต่อการหาค่าคำตอบ

สำหรับปัญหาขนาดกลาง ปัญหาที่ 1 และปัญหาที่ 2 ยกเว้น Iterated Local Search ของปัญหาที่ 2 จะมีค่า P – Value ที่น้อยกว่าระดับนัยสำคัญที่เท่ากับ 0.05 แสดงว่าสามารถปฏิเสธสมมติฐานหลัก และยอมรับสมมติฐานรอง ซึ่งก็คือ วิธีการปรับปรุงคำตอบทั้งแบบสลับ 1 แบบสลับ 2 และแบบเลื่อนที่ใช้นั้น มีผลต่อการหาค่าคำตอบ

ปัญหาขนาดใหญ่ ทั้งปัญหาที่ 1 และปัญหาที่ 2 แสดงให้เห็นว่าค่า P – Value มีค่าน้อยกว่าระดับนัยสำคัญที่เท่ากับ 0.05 แสดงว่าสามารถปฏิเสธสมมติฐานหลัก และยอมรับสมมติฐานรอง ซึ่งก็คือ วิธีการปรับปรุงคำตอบทั้งแบบสลับ 1 แบบสลับ 2 และแบบเลื่อนที่ใช้นั้น มีผลต่อการหาค่าคำตอบ

โดยวิธีการที่สามารถหาค่าคำตอบได้ดีของปัญหาขนาดใหญ่ ทั้งปัญหาที่ 1 และปัญหาที่ 2 คือวิธีการปรับปรุงคำตอบแบบสลับ 1 ซึ่งสามารถหาค่าระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะได้น้อยกว่าวิธีการปรับปรุงคำตอบแบบสลับ 2 และแบบ

จะเห็นได้ว่า โดยรวมแล้วจะพบวิธีการปรับปรุงคำตอบที่จะส่งผลต่อผลลัพธ์ของแต่ละวิธีการในการค้นหาคำตอบ

4.7.4 การวิเคราะห์ข้อมูลโดยการทดสอบแบบ t-test

กระบวนการทางสถิติ t-test เป็นการแจกแจกแบบ Student's สำหรับเปรียบเทียบค่าเฉลี่ย 2 ค่า นอกจากนั้นยังแสดงค่าเฉลี่ย ส่วนเบี่ยงเบนมาตรฐานและความคลาดเคลื่อนมาตรฐานในแต่ละตัวแปรด้วย ซึ่งสถิติ t-test สามารถแบ่งออกเป็น 3 กลุ่มด้วยกัน คือ

4.7.4.1 One-sample test คือ การเปรียบเทียบค่าเฉลี่ยของกลุ่มตัวอย่างกับค่าคงที่ใดๆ หรือการนำค่าเฉลี่ยของกลุ่มตัวอย่างมาเปรียบเทียบกับเกณฑ์ เช่น ต้องการทดสอบว่าค่าเฉลี่ยความพึงพอใจของกลุ่มตัวอย่างจะมีค่าเท่ากับ 3.50 หรือไม่

4.7.4.2 Independent t-test คือ การเปรียบเทียบค่าเฉลี่ยของกลุ่มตัวอย่างสองกลุ่มที่เป็นอิสระจากกัน เช่น ต้องการทดสอบว่าค่าเฉลี่ยของคะแนนสอบของนักเรียนชายและหญิงจะแตกต่างกันหรือไม่

4.7.4.3 Paired-sample test คือ การเปรียบเทียบค่าเฉลี่ยของกลุ่มตัวอย่าง 1 กลุ่ม ที่มีค่าเฉลี่ยให้เปรียบเทียบกัน 2 ค่า พูดกันง่ายๆ เช่น เอาค่าเฉลี่ยของคะแนนสอบก่อนเรียนกับหลังเรียนของนักเรียนมาเปรียบเทียบกันนั่นเอง

4.7.5 กระบวนการทดสอบสมมติฐาน

กระบวนการทดสอบสมมติฐานจะช่วยผู้วิจัยในการตัดสินใจสรุปผลว่ามีความสัมพันธ์กันระหว่างตัวแปรจริงหรือไม่หรือช่วยใจการตัดสินใจเพื่อสรุปผลว่าสิ่งที่นำมาเปรียบเทียบกันนั้นแตกต่างกันจริงหรือไม่

การทดสอบสมมติฐานประกอบด้วย 2 ส่วนคือ สมมติฐานหลัก และสมมติฐานรอง สำหรับโครงงานวิจัยนี้สมมติฐานหลัก คือ วิธีการปรับปรุงคำตอบไม่มีผลต่อการหาค่าคำตอบ และสมมติฐานรอง คือ วิธีการปรับปรุงคำตอบไม่มีผลต่อการหาค่าคำตอบ สำหรับการทดสอบสมมติฐาน

นั้นอยู่บนพื้นฐานของความน่าจะเป็นทางสถิติหรือเราเรียกว่าค่า P – Value ที่ได้จากการทดสอบเช่นเดียวกับการวิเคราะห์ความแปรปรวน

ก. ถ้าค่า P – Value มีค่าน้อยกว่าหรือเท่ากับระดับนัยสำคัญ (α - Level) ซึ่งเท่ากับ 0.05 ให้ปฏิเสธสมมติฐานหลักและยอมรับสมมติฐานรอง ซึ่งก็คือ วิธีการปรับปรุงคำตอบมีผลต่อการหาค่าคำตอบ

ข. ถ้าค่า P – Value มีค่ามากกว่าระดับนัยสำคัญ (α - Level) ซึ่งเท่ากับ 0.05 แสดงว่าให้ยอมรับสมมติฐานหลักและให้ปฏิเสธสมมติฐานรอง ซึ่งก็คือ วิธีการปรับปรุงคำตอบไม่มีผลต่อการหาค่าคำตอบ

4.7.6 สรุปผลการวิเคราะห์ข้อมูลโดยการทดสอบแบบ t-test ของการแก้ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก

การวิเคราะห์ข้อมูลโดยการทดสอบแบบ t-test มีค่าความน่าจะเป็นทางสถิติของโจทย์ปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุซึ่งมีขนาดปัญหา 3 ขนาด ได้แก่ ปัญหาขนาดเล็ก มีลูกค้า 10 ราย, ปัญหาขนาดกลาง มีลูกค้า 25 ราย และปัญหาขนาดใหญ่ มีลูกค้า 50 ราย แสดงไว้ดังตารางที่ 4.10 และกำหนดให้ระดับนัยสำคัญ มีค่าเท่ากับ 0.05

ตารางที่ 4.10 สรุปผลการทดสอบ t-test

โจทย์ปัญหา	วิธีการค้นหาคำตอบ	Swap1-Swap2	Swap1-Slide	Swap2-Slide
1. ปัญหาขนาดเล็ก โจทย์ปัญหาที่ 1	Local Search 1	0.009	0.000	0.013
	Local Search 2	0.187	0.042	0.008
	Iterated Local Search	0.094	0.019	0.001
2. ปัญหาขนาดเล็ก โจทย์ปัญหาที่ 2	Local Search 1	0.037	0.966	0.057
	Local Search 2	0.042	0.013	0.006
	Iterated Local Search	0.306	0.765	0.032
3. ปัญหาขนาดกลาง โจทย์ปัญหาที่ 1	Local Search 1	0.000	0.003	0.001
	Local Search 2	0.009	0.23	0.005
	Iterated Local Search	0.001	0.000	0.003

ตารางที่ 4.10 (ต่อ) สรุปผลการทดสอบ t-test

โจทย์ปัญหา	วิธีการค้นหาคำตอบ	Swap1-Swap2	Swap1-Slide	Swap2-Slide
4. ปัญหาขนาดกลาง โจทย์ปัญหาที่ 2	Local Search 1	0.000	0.000	0.001
	Local Search 2	0.091	0.016	0.000
	Iterated Local Search	0.028	0.002	0.481
5. ปัญหาขนาดใหญ่ โจทย์ปัญหาที่ 1	Local Search 1	0.000	0.000	0.487
	Local Search 2	0.005	0.000	0.000
	Iterated Local Search	0.001	0.000	0.003
6. ปัญหาขนาดใหญ่ โจทย์ปัญหาที่ 2	Local Search 1	0.000	0.000	0.112
	Local Search 2	0.075	0.000	0.000
	Iterated Local Search	0.000	0.000	0.000

จากตารางที่ 4.10 ซึ่งแสดงผลการทดสอบของ t-test พบว่า โดยส่วนใหญ่แล้วค่า P-Value จะมีค่าน้อยกว่าหรือเท่ากับระดับนัยสำคัญที่เท่ากับ 0.05 ซึ่งจะให้ปฏิเสธสมมติฐานหลักและยอมรับสมมติฐานรอง นั่นหมายความว่า วิธีการปรับปรุงคำตอบมีผลต่อการหาค่าคำตอบ และมีเพียงบางส่วนเท่านั้นที่วิธีการปรับปรุงคำตอบไม่มีผลต่อการหาค่าคำตอบ คือ มีค่า P-Value มากกว่าระดับนัยสำคัญที่เท่ากับ 0.05

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

5.1.1 จากการวิจัยในครั้งนี้ทำให้เกิดโปรแกรมสำเร็จรูปที่ช่วยในการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก เพื่อหาระยะทางการเดินทางของยานพาหนะที่สั้นที่สุดในการขนส่งสินค้าให้กับลูกค้า ซึ่งสามารถนำไปใช้ในการจัดเส้นทางการเดินทางสำหรับยานพาหนะในการขนส่งสินค้าให้กับลูกค้าแบบมีข้อจำกัดเนื่องจากความจุ

5.1.2 โปรแกรมการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติกจะทำการประมวลผลให้ได้คำตอบในการหาระยะทางการขนส่งสินค้าให้กับลูกค้าของยานพาหนะในเวลาที่รวดเร็ว และได้คำตอบเป็นที่น่าพึงพอใจ แม้คำตอบที่ได้นั้นอาจจะไม่ใช่คำตอบที่ดีที่สุดก็ตาม

5.1.3 จากการทดสอบประสิทธิภาพของโปรแกรมการหาคำตอบของปัญหาการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติกจำลองนี้ พบว่าสำหรับในปัญหาขนาดเล็ก เมื่อพิจารณาค่าระยะทางการขนส่งสินค้าให้กับลูกค้าของยานพาหนะเฉลี่ยนั้น พบว่า หลักการแบบสลับ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 สามารถให้คำตอบที่ดีกว่าหลักการแบบสลับ 2 และหลักการแบบเลื่อน ในปัญหาที่ 1 หากพิจารณาในด้านเวลาเฉลี่ยในการประมวลผลนั้น หลักการแบบสลับ 1 ใช้เวลาเฉลี่ยในการประมวลผลน้อยกว่า ส่วนหลักการแบบสลับ 2 และหลักการแบบเลื่อนนั้นจะมีเวลาเฉลี่ยในการประมวลผลที่ใกล้เคียงกันซึ่งมากกว่าหลักการแบบสลับ 1 เนื่องจากตัวโปรแกรมของหลักการแบบสลับ 2 และแบบเลื่อนนั้นมีความซับซ้อนมากกว่าหลักการแบบอื่น

ปัญหาขนาดเล็กของปัญหาที่ 2 เมื่อพิจารณาค่าระยะทางการขนส่งสินค้าให้กับลูกค้าของยานพาหนะเฉลี่ยนั้น พบว่า หลักการแบบเลื่อน โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 สามารถให้คำตอบที่ดีกว่าหลักการแบบสลับ 1 และหลักการแบบสลับ 2 แต่ยังคงให้ค่าระยะทางที่ใกล้เคียงกับหลักการแบบสลับ 1 มาก หากพิจารณาในด้านเวลาเฉลี่ยในการประมวลผลนั้น หลักการแบบเลื่อนจะใช้เวลาเฉลี่ยในการประมวลผลที่มากกว่า รองลงมาเป็นหลักการแบบสลับ 2 และหลักการแบบสลับ 1 ตามลำดับจากมากไปน้อย ซึ่งเนื่องจากกระบวนการในการค้นหาคำตอบของหลักการแบบเลื่อนนั้นมีความหลากหลายในการค้นหาคำตอบมากกว่าทั้ง 2 แบบ จึงทำให้มีโอกาสที่จะเจอค่าคำตอบที่ดีกว่าแบบอื่น แต่เนื่องด้วยตัวโปรแกรมมีความซับซ้อนมากกว่าจึงทำให้ใช้เวลาในการประมวลผลที่มากกว่าแบบอื่น

ปัญหาขนาดกลางของปัญหาที่ 1 หลักการแบบสลับ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2 มีค่าระยะทางการขนส่งสินค้าให้กับลูกค้าของยานพาหนะเฉลี่ยที่ดีกว่าหลักการแบบสลับ 1 และหลักการเลื่อน ซึ่งหลักการแบบเลื่อนจะให้ค่าระยะทางเฉลี่ยที่มากที่สุด หากพิจารณาในด้านเวลาเฉลี่ยในการประมวลผลนั้น หลักการแบบสลับ 1 จะมีเวลาเฉลี่ยใกล้เคียงกับหลักการแบบเลื่อนซึ่งใช้เวลาในการประมวลผลน้อยกว่าหลักการแบบเลื่อน

ปัญหาขนาดกลางของปัญหาที่ 2 หลักการแบบสลับ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 มีค่าระยะทางการขนส่งสินค้าให้กับลูกค้าของยานพาหนะเฉลี่ยที่ดีกว่าแบบสลับ 2 และหลักการเลื่อน หากพิจารณาในด้านเวลาเฉลี่ยในการประมวลผลนั้น หลักการแบบสลับ 1 ใช้เวลาเฉลี่ยในการประมวลผลน้อยกว่าหลักการแบบสลับ 2 และหลักการแบบเลื่อน

ปัญหาขนาดใหญ่ พบว่า การกระจายตัวของระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะมีมากขึ้น เนื่องจากขนาดของโจทย์ที่ใหญ่ขึ้น สำหรับปัญหาที่ 1 พบว่า หลักการแบบสลับ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 ให้ค่าระยะทางในการขนส่งสินค้าให้กับลูกค้าของยานพาหนะเฉลี่ยน้อยกว่าหลักการแบบสลับ 2 และหลักการแบบเลื่อน หากพิจารณาในด้านเวลาเฉลี่ยที่ใช้ในการประมวลผลนั้น หลักการแบบสลับ 1 จะใช้เวลาในการประมวลผลน้อยกว่าหลักการอื่น ส่วนหลักการที่ใช้เวลาในการประมวลผลรองลงมา คือ หลักการแบบสลับ 2 และหลักการแบบเลื่อน ตามลำดับ

ปัญหาขนาดใหญ่ของปัญหาที่ 2 พบว่า เมื่อพิจารณาค่าระยะทางการขนส่งสินค้าให้กับลูกค้าของยานพาหนะเฉลี่ยนั้น หลักการแบบสลับ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2 มีระยะทางเฉลี่ยน้อยที่สุด รองลงมา คือ หลักการแบบสลับ 2 และหลักการแบบเลื่อน ตามลำดับ เมื่อพิจารณาระยะเวลาที่ใช้ในการประมวลผล พบว่า หลักการแบบสลับ 2 มีระยะเวลาเฉลี่ยน้อยที่สุด รองลงมา คือ หลักการแบบสลับ 1 และหลักการแบบเลื่อน

จากการพิจารณาปัญหาทั้ง 3 ขนาด จะพบว่า หากต้องการหาคำตอบในย่านปัญหาขนาดเล็กซึ่งมีจำนวนยานพาหนะ 3 คัน และจำนวนลูกค้า 10 ราย ควรใช้หลักการปรับปรุงคำตอบแบบสลับ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 แต่ก็ไม่ควรละทิ้งหลักการแบบเลื่อน เนื่องจากหลักการแบบเลื่อน โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 มีโอกาสให้คำตอบที่ดีกว่าแบบสลับได้เช่นกัน ส่วนคำตอบในย่านของปัญหาขนาดกลางซึ่งมีจำนวนยานพาหนะ 7 คัน และจำนวนลูกค้า 25 ราย ควรใช้หลักการปรับปรุงคำตอบแบบสลับ 2 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 2 เนื่องจากเป็นหลักการที่ให้ค่าคำตอบที่ดีกว่าแบบอื่น และสำหรับคำตอบในย่านปัญหาขนาดใหญ่ ซึ่งมีจำนวนยานพาหนะ 13 คัน และจำนวนลูกค้า 50 ราย หลักการปรับปรุงคำตอบแบบสลับ 1 โดยการค้นหาคำตอบเฉพาะที่ของ Local Search 1 เป็นหลักการที่ควรนำมาใช้ในการหาคำตอบมากกว่าหลักการแบบอื่น เนื่องจากเป็นหลักการที่ให้ค่าคำตอบและเวลาที่ใช้ในการประมวลผลที่ดีกว่าแบบอื่น

5.1.4 การวิเคราะห์การปรับปรุงคำตอบ ซึ่งจากการทดสอบโปรแกรมนี้ พบว่า ส่วนใหญ่วิธีการปรับปรุงคำตอบจะส่งผลต่อผลลัพธ์ของวิธีการปรับปรุงคำตอบ นั่นก็คือ มีค่า P – Value ที่น้อยกว่าระดับนัยสำคัญที่เท่ากับ 0.05 ซึ่งแสดงว่าสามารถปฏิเสธสมมติฐานหลัก และยอมรับสมมติฐานรอง ซึ่งก็คือ วิธีการปรับปรุงคำตอบทั้งแบบสลับ 1 แบบสลับ 2 และแบบเลื่อนที่ใช้นั้น มีผลต่อการหาค่าคำตอบ ซึ่งได้แก่ ปัญหาขนาดเล็ก โจทย์ปัญหาที่ 1 ของ Local Search 1 แบบสลับ 1 ปัญหาขนาดเล็ก โจทย์ปัญหาที่ 2 Local Search 2 แบบสลับ 2 ปัญหาขนาดกลาง โจทย์ปัญหาที่ 1 และโจทย์ปัญหาที่ 2 ของ Local Search 1 และLocal Search 2 แบบสลับ 1 รวมทั้งโจทย์ปัญหาที่ 1 ของ Iterated Local Search แบบสลับ 1 และปัญหาขนาดใหญ่ทั้งโจทย์ปัญหาที่ 1 และโจทย์ปัญหาที่ 2 ของ Local Search 1 Local Search 2 และ Iterated Local Search แบบสลับ 1 จะส่งผลต่อการปรับปรุงคำตอบ

5.2 ปัญหาในการดำเนินโครงงานวิจัย

5.2.1 เนื่องจากผู้วิจัยไม่มีความรู้ในการเขียนโปรแกรม Visual Basic for Applications จึงทำให้ระยะแรกๆ ในการจัดทำโครงงานค่อนข้างดำเนินไปด้วยความล่าช้า

5.2.2 การเขียนโปรแกรมและทดสอบโปรแกรมต้องใช้เวลามาก

5.3 ข้อเสนอแนะ

5.3.1 สำหรับจำนวนรอบในการค้นหาคำตอบนั้น อาจเพิ่มจำนวนรอบในการค้นหาคำตอบได้ เพราะยิ่งถ้าจำนวนรอบในการค้นหาคำตอบมากขึ้นก็จะทำให้สามารถเจอค่าคำตอบที่ดีที่สุดขึ้นได้

5.3.2 ในการดำเนินการปรับปรุงคำตอบนั้น อาจจะรูปแบบมีความหลากหลายที่จะให้คำตอบที่ดีกว่า สะดวกกว่า และรวดเร็วกว่าในการประมวลผล ซึ่งนั่นอาจจะเป็นเหตุให้จะต้องมีการศึกษาในการการปรับปรุงคำตอบหลายๆ แบบต่อไป

5.3.3 โปรแกรมนี้จะสามารถนำไปให้ได้กับกรณีที่ใช้ในการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุของสินค้าที่ลูกค้าต้องการซึ่งความจุนั้นจะต้องไม่เกินความจุสูงสุดของยานพาหนะเท่านั้น หากผู้ใช้จะนำไปใช้กับการจัดเส้นสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากปัจจัยอื่นๆ สามารถนำโปรแกรมนี้ไปพัฒนาเพื่อนำไปประยุกต์ใช้กับปัญหาของท่านได้

5.3.4 ผู้ใช้ที่นำโปรแกรมการจัดเส้นทางสำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยวิธีเมตาฮิวริสติกนี้ไปใช้ ควรตรวจสอบก่อนว่าจำนวนความต้องการสินค้ารวมของลูกค้าทั้งหมดเพียงพอต่อจำนวนยานพาหนะหรือไม่

เอกสารอ้างอิง

ชวลิตร พันธุ์ชมภู. (2550). การขนส่งผลิตภัณฑ์น้ำแข็ง. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาการจัดการงานวิศวกรรม มหาวิทยาลัยศิลปากร.

ดุสิต กอปรรักชาติ. (2554). Advanced Excel ฉบับเขียนโปรแกรมด้วย Macro & VBA. กรุงเทพฯ : โปรวิชั่น

ธรินี มณีศรี. (2550). การพัฒนาอัลกอรีทึมสำหรับปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีหลายคลังสินค้าและมีความไม่แน่นอนภายใต้กรอบเวลา. ปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต สาขาวิศวกรรมอุตสาหการ มหาวิทยาลัยเกษตรศาสตร์

ธรินี มณีศรี. (2552). ขั้นตอนวิธีการสำหรับการหาผลเฉลยเชิงทนทานของปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลาและเวลาเดินทางไม่แน่นอน. วิทยานิพนธ์วิศวกรรมศาสตรดุษฎีบัณฑิต สาขาวิศวกรรมอุตสาหการ มหาวิทยาลัยเกษตรศาสตร์.

มาลินี หลวงคลัง และคณะ. (2554). โปรแกรมช่วยจัดเส้นทางโดยใช้วิธีเจนเนติกอัลกอริทึม. ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหการ มหาวิทยาลัยนเรศวร.

ระพีพันธ์ ปิตาคะโส. (2554). วิธีการเมตาฮิวริสติกเพื่อแก้ไขปัญหาการวางแผนการผลิตและการจัดการโลจิสติกส์. กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท. สมาคมส่งเสริมเทคโนโลยี (ไทย – ญี่ปุ่น).

สมชาย ปฐมศิริ. (กันยายน 2551). โครงการพัฒนาหลักสูตรและการฝึกอบรมโลจิสติกส์และซัพพลายเชน. สืบค้นเมื่อวันที่ 1 กรกฎาคม 2555, จาก http://www.chiangrailogistics.com/files/C1_Intro.ppt

Olivier C. and et al. Handbook on MetaHeuristics. Ed. F. Glover and G. Kochenberger. สืบค้นเมื่อวันที่ 7 สิงหาคม 2555, จาก http://sci2s.ugr.es/docencia/metaheuristicas/ILS.pdf

R. Battiti and et al. Reactive Search and Intelligent Optimization. สืบค้นเมื่อวันที่ 4 สิงหาคม 2555, จาก http://lion.disi.unitn.it/reactive – searh/thebook/node10. Html.

ภาคผนวก

Source Code ของโปรแกรมการหาคำตอบของปัญหาการจัดเส้นทาง สำหรับยานพาหนะแบบมีข้อจำกัดเนื่องจากความจุโดยใช้วิธีเมตาฮิวริสติก

เพื่อสะดวกแก่ความเข้าใจ จะขอแบ่งการแสดงคำสั่งหรือ Source Code ตามลักษณะของโปรแกรม ซึ่งจะแบ่งได้ดังนี้

1.การเปิดโปรแกรมและหน้าแรกที่เป็นส่วนต้อนรับโปรแกรม

2. โค้ดการทำงานของ User form Data

3. โค้ดการทำงานของ Worksheets Data

4. โค้ดการทำงานของ Worksheets Run of Cycle

5. โค้ดการทำงานของ Module 1

1. การเปิดโปรแกรมและหน้าแรกที่เป็นส่วนต้อนรับโปรแกรม

เมื่อเปิดโปรมแกรมจะเข้าหน้าแรกที่เป็นต้อนรับของโปรแกรมโดยอัตโนมัติ

```
Private Sub Workbook_Open()
    Wellcome.Show
End Sub
```

รูปที่ 1 หน้าแรกที่เป็นต้อนรับของโปรแกรมโดยอัตโนมัติ

จากรูปที่ 4.11 ของบทที่ 4 ในส่วนหน้าแรกของโปรแกรมจะมีปุ่มให้กดอยู่ 3 ปุ่ม จะแสดงคำสั่งเมื่อกดปุ่มหมายเลข 1, 2, 3 ตามลำดับ

```
Private Sub Start1_Click()
    Worksheets("Data").Activate
    Data.Show
    Wellcome.Hide
End Sub
```

รูปที่ 2 แสดงคำสั่งเมื่อคลิ้กปุ่ม START ในหน้าแรกของโปรแกรม

```
Private Sub Help1_Click()
    Help.Show
End Sub
```

รูปที่ 3 แสดงคำสั่งเมื่อคลิ้ก HELP ในหน้าแรกของโปรแกรม

```
Private Sub Exit1_Click()
   ConfirmExit.Show
End Sub
```

รูปที่ 4 แสดงคำสั่งเมื่อคลิ้ก EXIT ในหน้าแรกของโปรแกรม

2. โค้ดการทำงานของ User form Data

```
.....ประกาศตัวแปร
Public Sub OK1_Click()
   Dim NumberVehicle As Integer
   Dim NumberCustomer As Integer
   Dim MaximumCapacity As Integer
   Dim NumberCycle As Integer
   Dim SubILS As Integer
   Dim i As Integer
```

รูปที่ 5 การทำงานของ User form Data

```
.....การแจ้งเตือนเมื่อกรอกข้อมูลเป็นทศนิยมของช่องจำนวนยานพาหนะ
 If IsNumeric(NumVehicle.Value) = True Then
       If (NumVehicle.Value / (Int(NumVehicle.Value))) <> 1 Then
         MsgBox "กรุณากรอกข้อมูลในช่องจำนวนยานพาหนะเป็นตัวเลขจำนวนเต็มเท่านั้น", vbInformation
          Exit Sub
        End If
End If
.....การแจ้งเตือนเมื่อไม่กรอกข้อมูลในช่องจำนวนยานพาหนะ
If NumVehicle.Value = "" Then
    MsgBox "กรุณากรอกข้อมูลตัวเลขในช่องจำนวนยานพาหนะ ", vbInformation
     Exit Sub
 End If
If IsNumeric(NumCustomer.Value) = True Then
    If (NumCustomer.Value / (Int(NumCustomer.Value))) <> 1 Then
       MsgBox "กรุณากรอกข้อมูลในช่องจำนวนลูกค้าเป็นตัวเลขจำนวนเต็มเท่านั้น", vbInformation
        Exit Sub
     End If
```

รูปที่ 5 (ต่อ) การทำงานของ User form Data

```
If NumCustomer.Value = "" Then
        MsgBox "กรุณากรอกข้อมูลลงในช่องจำนวนลูกค้า", vbInformation
        Exit Sub
    End If

 If IsNumeric(MaxCapacity.Value) = True Then
     If (MaxCapacity.Value / (Int(MaxCapacity.Value))) <> 1 Then
         MsgBox "กรุณากรอกข้อมูลในช่องความจุสูงสุดของยานพาหนะเป็นตัวเลขจำนวนเต็มเท่านั้น", vbInformation
          Exit Sub
       End If
If MaxCapacity.Value = "" Then
        MsgBox "กรุณากรอกข้อมูลตัวเลขในช่องความจุสูงสุดของยานพาหนะ", vbInformation
        Exit Sub
 End If

If IsNumeric(NumCycle.Value) = True Then
    If (NumCycle.Value / (Int(NumCycle.Value))) <> 1 Then
        MsgBox "กรุณากรอกข้อมูลในช่องจำนวนรอบในการหาคำตอบเป็นตัวเลขจำนวนเต็มเท่านั้น", vbInformation
          Exit Sub
       End If
  End If
 If NumCycle.Value = "" Then
     MsgBox "กรุณากรอกข้อมูลตัวเลขในช่องจำนวนรอบในการค้นหาคำตอบ", vbInformation
     Exit Sub
 End If
```

รูปที่ 5 (ต่อ) การทำงานของ User form Data

```
.....การแจ้งเตือน ถ้าไม่เลือกวิธีการค้นหาคำตอบที่ดีที่สุด
If LoSearch1.Value = False And LoSearch2.Value = False And ItLoSearch.Value = False Then
        MsgBox "กรุณาเลือกวิธีที่จะใช้ในการค้นหาคำตอบ Local Search 1 หรือ Local Search 2 หรือ
Iterated Local Search", vbInformation
    Exit Sub
End If
.....การแจ้งเตือน ถ้าไม่เลือกวิธีการค้นหาคำตอบที่ดีที่สุด
If LoSearch1.Value = False And LoSearch2.Value = False And ItLoSearch.Value = False Then
        MsgBox "กรุณาเลือกวิธีที่จะใช้ในการค้นหาคำตอบ Local Search 1 หรือ Local Search 2 หรือ
Iterated Local Search", vbInformation
    Exit Sub
End If
```

รูปที่ 6 การแจ้งเตือน

```
.....การแจ้งเตือน เมื่อไม่เลือกวิธีการในการปรับปรุงคำตอบ
If LoSearch1.Value = True And Swap1.Value = False And Swap2.Value = False And
 Slide1.Value = False Then
        MsgBox "กรุณาเลือกวิธีที่ใช้ในการปรับปรุงคำตอบของ Local Search1", vbInformation
        Exit Sub
 End If
If LoSearch2.Value = True And Swap3.Value = False And Swap4.Value = False And
 Slide2.Value = False Then
        MsgBox "กรุณาเลือกวิธีที่ใช้ในการปรับปรุงคำตอบของ Local Search2", vbInformation
        Exit Sub
 End If
If ItLoSearch.Value = True And Swap5.Value = False And Swap6.Value = False And
 Slide3.Value = False Then
        MsgBox "กรุณาเลือกวิธีที่ใช้ในการปรับปรุงคำตอบของ Iterated Local Search",
 vbInformation
        Exit Sub
End If
```

รูปที่ 6 (ต่อ) การแจ้งเตือน

```
.....การแจ้งเตือน เมื่อไม่กรอกจำนวนรอบในการวนซ้ำของ Iterated Local Search
If Swap5.Value = True And SubCycle.Value = "" Then
    MsgBox "กรุณาระบุจำนวนรอบการวนซ้ำ ในการค้นหาคำตอบแบบ Iterated Local Search",
  vbInformation
    Exit Sub
End If
If Swap6.Value = True And SubCycle.Value = "" Then
    MsgBox " กรุณาระบุจำนวนรอบการวนซ้ำ ในการค้นหาคำตอบแบบ Iterated Local Search",
  vbInformation
    Exit Sub
End If
If Slide3.Value = True And SubCycle.Value = "" Then
    MsgBox "กรุณาระบุจำนวนรอบการวนซ้ำ ในการค้นหาคำตอบแบบ Iterated Local Search",
 vbInformation
    Exit Sub
End If
If IsNumeric(SubCycle.Value) = True Then
    If (SubCycle.Value / (Int(SubCycle.Value))) <> 1 Then
      MsgBox "กรุณากรอกข้อมูลในช่องจำนวนรอบการวนซ้ำของ ILS เป็นตัวเลขจำนวนเต็มเท่านั้น",
  vbInformation
                Exit Sub
```

รูปที่ 6 (ต่อ) การแจ้งเตือน

```
.....การแจ้งเตือน เมื่อกรอกข้อมูลเป็นตัวหนังสือ
Dim A As Variant
 A = NumVehicle.Value
  If Len(A) > 0 Then
     If IsNumeric(A) = False Then
       MsgBox "กรุณากรอกข้อมูลจำนวนยานพาหนะเป็นตัวเลขเท่านั้น", vbInformation
        Exit Sub
      End If
   End If
   NumberVehicle = NumVehicle.Value  'ให้แสดงค่าใน User form Data บนช่อง Excel
   Worksheets("Data").Range("K17") = NumVehicle.Value
   Worksheets("Data").Range("K17").Interior.Color = vbYellow 'ใส่สีช่อง Excel
   Worksheets("Data").Range("K17").Font.Size = 14  'กำหนดขนาดตัวหนังสือ บนช่อง Excel
   Worksheets("Run of Cycle").Range("C17") = NumVehicle.Value
   Worksheets("Run of Cycle").Range("C17").Interior.Color = vbYellow
   Worksheets("Run of Cycle").Range("C17").Font.Size = 14
   Worksheets("Run of Cycle").Range("C17").Font.Bold = True

 Dim b As Variant
 b = NumCustomer.Value
  If Len(b) > 0 Then
       If IsNumeric(b) = False Then
          MsgBox "กรุณากรอกข้อมูลจำนวนลูกค้าเป็นตัวเลขเท่านั้น", vbInformation
          Exit Sub
      End If
    End If
   NumberCustomer = NumCustomer.Value 'ให้แสดงค่าใน User form Data บนช่อง Excel
    Worksheets("Data").Range("K18") = NumberCustomer
    Worksheets("Data").Range("K18").Interior.Color = vbYellow
    Worksheets("Run of Cycle").Range("C18") = NumCustomer.Value
    Worksheets("Run of Cycle").Range("C18").Interior.Color = vbYellow
     Worksheets("Run of Cycle").Range("C18").Font.Size = 14
     Worksheets("Run of Cycle").Range("C18").Font.Bold = True
```

รูปที่ 6 (ต่อ) การแจ้งเตือน

```
Dim c As Variant
 c = MaxCapacity.Value
 If Len(c) > 0 Then
        If IsNumeric(c) = False Then
            MsgBox "กรุณากรอกข้อมูลความจุสูงสุดของยานพาหนะเป็นตัวเลขเท่านั้น",
    vbInformation
            Exit Sub
       End If
     End If
 MaximumCapacity = MaxCapacity.Value
        Worksheets("Data").Range("K19") = MaximumCapacity
        Worksheets("Data").Range("K19").Interior.Color = vbYellow
        Worksheets("Data").Range("K19").Font.Size = 14

Dim d As Variant
d = NumCycle.Value
 If Len(d) > 0 Then
     If IsNumeric(d) = False Then
            MsgBox "กรุณากรอกข้อมูลจำนวนรอบในการค้นหาคำตอบเป็นตัวเลขเท่านั้น",
    vbInformation
            Exit Sub
       End If
     End If
NumberCycle = NumCycle.Value
        Worksheets("Data").Range("K21") = NumberCycle
        Worksheets("Data").Range("K21").Interior.Color = vbYellow
        Worksheets("Data").Range("K21").Font.Size = 14
        Worksheets("Run of Cycle").Range("C20") = NumCycle.Value
        Worksheets("Run of Cycle").Range("C20").Interior.Color = vbYellow
        Worksheets("Run of Cycle").Range("C20").Font.Size = 16
        Worksheets("Run of Cycle").Range("C20").Font.Bold = True
```

รูปที่ 6 (ต่อ) การแจ้งเตือน

```
.....การจัดกลางตัวหนังสือบน Excel
NumberCustomer = Worksheets("Data").Range("K18").Value
  For i = 1 To NumberCustomer
     For j = 1 To NumberCustomer + 1
        Worksheets("Data").Range("B27").Offset(i, 0).HorizontalAlignment = xlCenter
        Worksheets("Data").Range("C27").Offset(i, 0).HorizontalAlignment = xlCenter
        Worksheets("Data").Range("G27").Offset(0, j).HorizontalAlignment = xlCenter
        Worksheets("Data").Range("G27").Offset(j, 0).HorizontalAlignment = xlCenter
        Worksheets("Data").Range("G27").Offset(j, j).HorizontalAlignment = xlCenter
        Worksheets("Data").Range("G28").Offset(0, j).HorizontalAlignment = xlCenter
        Worksheets("Data").Range("G28").Offset(i, j).HorizontalAlignment = xlCenter
     Next j
  Next i

.....การแสดงค่าจากการเลือกวิธีการค้นหาคำตอบที่ดีที่สุด
  '** แสดงการสลับ 1
  If Swap1.Value = True Or Swap3.Value = True Or Swap5.Value = True Then
     Worksheets("Run of Cycle").Range("C22").Value = "แบบสลับ 1"
     Worksheets("Run of Cycle").Range("C22").Font.Size = 14
     Worksheets("Run of Cycle").Range("H12").Value = "แบบสลับ 1"
     Worksheets("Run of Cycle").Range("H12").Interior.ColorIndex = 33
     Worksheets("Run of Cycle").Range("H12").Font.Size = 14
     Worksheets("Run of Cycle").Range("H12").HorizontalAlignment = xlCenter
  End If
  '**แสดงการสลับ 2
   If Swap2.Value = True Or Swap4.Value = True Or Swap6.Value = True Then
     Worksheets("Run of Cycle").Range("C22").Value = "แบบสลับ 2"
     Worksheets("Run of Cycle").Range("C22").Font.Size = 14
     Worksheets("Run of Cycle").Range("H12").Value = "แบบสลับ 2"
     Worksheets("Run of Cycle").Range("H12").Interior.ColorIndex = 33
     Worksheets("Run of Cycle").Range("H12").Font.Size = 14
     Worksheets("Run of Cycle").Range("H12").HorizontalAlignment = xlCenter
  End If
```

รูปที่ 7 การจัดกลางตัวหนังสือบน Excel

```
'**แสดงการเลื่อน
    If Slide1.Value = True Or Slide2.Value = True Or Slide3.Value = True Then
        Worksheets("Run of Cycle").Range("C22").Value = "แบบเลื่อน"
        Worksheets("Run of Cycle").Range("C22").Font.Size = 14
        Worksheets("Run of Cycle").Range("H12").Value = "แบบเลื่อน"
        Worksheets("Run of Cycle").Range("H12").Interior.ColorIndex = 33
        Worksheets("Run of Cycle").Range("H12").Interior.ColorIndex = 33
        Worksheets("Run of Cycle").Range("H12").Font.Size = 14
        Worksheets("Run of Cycle").Range("H12").HorizontalAlignment = xlCenter
    End If
    If LoSearch1.Value = True Then
        Worksheets("Run of Cycle").Range("F12").Value = "Local Search 1"
        Worksheets("Run of Cycle").Range("F12").Interior.ColorIndex = 4
        Worksheets("Run of Cycle").Range("F12").Font.Size = 14
        Worksheets("Run of Cycle").Range("F12").HorizontalAlignment = xlCenter
    End If
If LoSearch2.Value = True Then
        Worksheets("Run of Cycle").Range("F12").Value = "Local Search 2"
        Worksheets("Run of Cycle").Range("F12").Interior.ColorIndex = 4
        Worksheets("Run of Cycle").Range("F12").Font.Size = 14
        Worksheets("Run of Cycle").Range("F12").HorizontalAlignment = xlCenter
    End If
    If ItLoSearch.Value = True Then
        Worksheets("Run of Cycle").Range("F12").Value = "Iterated Local Search"
        Worksheets("Run of Cycle").Range("F12").Interior.ColorIndex = 4
        Worksheets("Run of Cycle").Range("F12").Font.Size = 14
        Worksheets("Run of Cycle").Range("F12").HorizontalAlignment = xlCenter
    End If
```

รูปที่ 8 แสดงการเลื่อน

```
    If ItLoSearch.Value = True Then
        Worksheets("Run of Cycle").Range("F12").Value = "Iterated Local Search"
        Worksheets("Run of Cycle").Range("F12").Interior.ColorIndex = 4
        Worksheets("Run of Cycle").Range("F12").Font.Size = 14
        Worksheets("Run of Cycle").Range("F12").HorizontalAlignment = xlCenter
    End If
    If SubCycle.Value <> "" Then
        SubILS = SubCycle.Value
        Worksheets("Run of Cycle").Range("C26") = SubILS
        Worksheets("Run of Cycle").Range("C26").Font.Size = 14
    Else: SubCycle.Value = ""
        Worksheets("Run of Cycle").Range("C26").Value = "-"
        Worksheets("Run of Cycle").Range("C26").Font.Size = 14
        Worksheets("Run of Cycle").Range("C26").HorizontalAlignment = xlCenter
    End If

    Application.ScreenUpdating = True

End Sub
```

รูปที่ 8 (ต่อ) แสดงการเลื่อน

3. โค้ดการทำงานของ Worksheets Data

```
Private Sub Check1_Click()
.....ประกาศตัวแปร
Dim i As Integer
Dim j As Integer

Worksheets("Data").Activate

Dim NumberVehicle As Integer
      If Range("K17").Value = "" Then
         MsgBox "กรุณากรอกข้อมูลลงในช่องจำนวนยานพาหนะ", vbInformation
         Exit Sub
      End If
If IsNumeric(Range("K17").Value) = True Then
    If (Range("K17").Value / (Int(Range("K17").Value))) <> 1 Then
       MsgBox "กรุณากรอกข้อมูลในช่องจำนวนยานพาหนะเป็นตัวเลขจำนวนเต็มเท่านั้น",
   vbInformation
        Exit Sub
    End If
End If
Dim A As Variant
  A = Range("K17").Value
  If Len(A) > 0 Then
     If IsNumeric(A) = False Then
        MsgBox "กรุณากรอกข้อมูลในช่องจำนวนยานพาหนะเป็นตัวเลขเท่านั้น", vbInformation
        Exit Sub
     End If
End If

Dim NumberCustomer As Integer
     If Range("CellCustomer").Value = "" Then
        MsgBox "กรุณากรอกข้อมูลลงในช่องจำนวนลูกค้า", vbInformation
        Exit Sub
     End If
```

รูปที่ 9 โค้ดการทำงานของ Worksheets Data

```
If IsNumeric(Range("CellCustomer").Value) = True Then
    If (Range("CellCustomer").Value / (Int(Range("CellCustomer").Value))) <> 1 Then
     MsgBox "กรุณากรอกข้อมูลในช่องจำนวนลูกค้าเป็นตัวเลขจำนวนเต็มเท่านั้น", vbInformation
      Exit Sub
    End If
End If

Dim b As Variant
  b = Range("K18").Value
  If Len(b) > 0 Then
    If IsNumeric(b) = False Then
      MsgBox "กรุณากรอกข้อมูลในช่องจำนวนลูกค้าเป็นตัวเลขเท่านั้น", vbInformation
       Exit Sub
    End If
  End If

Dim MaximumCapacity As Integer
    If Range("K19").Value = "" Then
      MsgBox "กรุณากรอกข้อมูลในช่องความจุสูงสุดของยานพาหนะ", vbInformation
      Exit Sub
    End If
Dim c As Variant
  c = Range("K19").Value
  If Len(c) > 0 Then
    If IsNumeric(c) = False Then
     MsgBox "กรุณากรอกข้อมูลในช่องความจุสูงสุดของยานพาหนะเป็นตัวเลขเท่านั้น",
   vbInformation
        Exit Sub
    End If
  End If
Dim NumberCycle As Integer
    If Range("K21").Value = "" Then
      MsgBox "กรุณากรอกข้อมูลในช่องจำนวนรอบในการค้นหาคำตอบ", vbInformation
      Exit Sub
    End If
```

รูปที่ 9 (ต่อ) โค้ดการทำงานของ Worksheets Data

```
Dim d As Variant
   d = Range("K21").Value
   If Len(d) > 0 Then
      If IsNumeric(d) = False Then
         MsgBox "กรุณากรอกข้อมูลในช่องจำนวนรอบในการค้นหาคำตอบเป็นตัวเลขเท่านั้น",
   vbInformation
         Exit Sub
      End If
   End If

   NumberCustomer = Worksheets("Data").Range("K18").Value
   For i = 1 To NumberCustomer
      If Range("C27").Offset(i, 0).Value = "" Then
         MsgBox "กรุณากรอกข้อมูลลงในช่องความต้องการสินค้าของลูกค้าจุดที่ " & i, vbInformation
         Exit Sub
      End If
    Next i
  NumberCustomer = Worksheets("Data").Range("K18")
  For i = 1 To NumberCustomer
  Dim e As Variant
   e = Range("C27").Offset(i, 0).Value
   If Len(e) > 0 Then
      If IsNumeric(e) = False Then
         MsgBox "กรุณากรอกข้อมูลลงในช่องความต้องการสินค้าของลูกค้าจุดที่ " & i & "เป็นตัวเลข
    เท่านั้น", vbInformation
         Exit Sub
      End If
   End If
   Next i
   For i = 1 To NumberCustomer
      For j = 1 To NumberCustomer + 1
         If Range("G27").Offset(i, j).Value = "" Then
            MsgBox "กรุณากรอกข้อมูลลงในช่องระยะทางของลูกค้าแต่ละจุดระหว่างลูกค้าคนที่ " & i &
     "และคนที่ " & j - 1, vbInformation
            Exit Sub
```

รูปที่ 9 (ต่อ) โค้ดการทำงานของ Worksheets Data

```
            End If
        Next j
    Next i

Dim f As Variant
 NumberCustomer = Worksheets("Data").Range("K18")
 f = Range("G27").Offset(i, j).Value

For i = 1 To NumberCustomer
    For j = 1 To NumberCustomer + 1
        If Len(f) > 0 Then
            If IsNumeric(f) = False Then
                MsgBox "กรุณากรอกข้อมูลลงในช่องระยะทางของลูกค้าแต่ละจุดระหว่างลูกค้าคนที่ " & i &
    "และคนที่ " & j - 1& "เป็นตัวเลขเท่านั้น" vbInformation
                Exit Sub
            End If
        End If
    Next j
    Next i

End Sub
```

รูปที่ 9 (ต่อ) โค้ดการทำงานของ Worksheets Data

```
.....การทำงานของปุ่ม Save
Private Sub Save1_Click()
    If MsgBox("คุณต้องการบันทึกข้อมูลทั้งหมดใช่หรือไม่", vbOKCancel, "Save") = vbCancel Then  Exit Sub
    ActivateWorkbook.Save
End Sub
```

รูปที่ 10 การทำงานของปุ่ม Save

```
.....การทำงานของปุ่ม Clear
Private Sub Clear_Click()
    ConfirmReset.Show
End Sub
```

รูปที่ 11 การทำงานของปุ่ม Clear

```
.....โค้ดของ User form ConfirmReset
'.....ถ้าคลิกปุ่ม OK
Private Sub OK_Click()
Dim j  As Integer
Dim i As Integer
Dim NumberCustomer As Integer
Dim NumberVehicle As Integer
Application.ScreenUpdating = False
ConfirmReset.Hide

NumberVehicle = Worksheets("Run of Cycle").Range("K17").Value
NumberCustomer = Worksheets("Data").Range("K18").Value
```

รูปที่ 12 โค้ดของ User form ConfirmReset

```
For j = 1 To NumberCustomer
   For i = 1 To NumberCustomer + 1
         Worksheets("Data").Range("K17:K19,K21").ClearContents
         Worksheets("Data").Range("B27").Offset(j, 0).ClearContents
         Worksheets("Data").Range("B27").Offset(j, 0).ClearFormats
         Worksheets("Data").Range("C27").Offset(j, 0).ClearContents
         Worksheets("Data").Range("C27").Offset(j, 0).ClearFormats
         Worksheets("Data").Range("G27").Offset(0, i).ClearContents
        Worksheets("Data").Range("G27").Offset(0, i).ClearFormats
         Worksheets("Data").Range("G27").Offset(i, 0).ClearContents
         Worksheets("Data").Range("G27").Offset(i, 0).ClearFormats
         Worksheets("Data").Range("H27").Offset(i, j).ClearContents
         Worksheets("Data").Range("H27").Offset(i, j).ClearFormats
         Worksheets("Data").Range("H27").Offset(i, 0).ClearContents
         Worksheets("Data").Range("H27").Offset(i, 0).ClearFormats
      Next i
   Next j

   Application.ScreenUpdating = True
End Sub
```

รูปที่ 12 (ต่อ) โค้ดของ User form ConfirmReset

```
'.....ถ้าคลิกปุ่ม Cancle
Private Sub Cancel_Click()
    Worksheets("Data").Activate
    ConfirmReset.Hide
End Sub
.....การทำงานของปุ่ม Run
Private Sub Run1_Click()
    Worksheets("Run of Cycle").Activate
End Sub
.....การทำงานของปุ่ม Back
Private Sub Back1_Click()
    Data.Show
End Sub
.....การทำงานของปุ่ม ยืนยันข้อมูลเบื้องต้นที่แก้ไข
Private Sub Confirm_Click()
    Dim CellCustomer As Integer
    Dim NumberCustomer As Integer
    Dim i As Integer
    Dim j As Integer
    Application.ScreenUpdating = False

If IsNumeric(Range("K17").Value) = True Then
    If (Range("K17").Value / (Int(Range("K17").Value))) <> 1 Then
        MsgBox "กรุณากรอกข้อมูลในช่องจำนวนยานพาหนะเป็นตัวเลขจำนวนเต็มเท่านั้น", vbInformation
        Exit Sub
    End If
End If
```

รูปที่ 12 (ต่อ) โค้ดของ User form ConfirmReset

```
If Range("K17").Value = "" Then
   MsgBox "กรุณากรอกข้อมูลในช่องจำนวนยานพาหนะ", vbInformation
    Exit Sub
End If

If IsNumeric(Range("K18").Value) = True Then
    If (Range("K18").Value / (Int(Range("K18").Value))) <> 1 Then
        MsgBox "กรุณากรอกข้อมูลในช่องจำนวนลูกค้าเป็นตัวเลขจำนวนเต็มเท่านั้น",
   vbInformation
         Exit Sub
      End If
 Else: Range("K18").Value.Value = ""
        MsgBox "กรุณากรอกข้อมูลในช่องจำนวนลูกค้า", vbInformation
         Exit Sub
   End If
'เรียกโค้ดสร้างตารางตามจำนวนลูกค้า
   Call TableSheetData

.....การจัดกลางตัวหนังสือ
 NumberCustomer = Worksheets("Data").Range("K18").Value
  For i = 1 To NumberCustomer
     For j = 1 To NumberCustomer + 1
         Worksheets("Data").Range("B27").Offset(i, 0).HorizontalAlignment = xlCenter
         Worksheets("Data").Range("C27").Offset(i, 0).HorizontalAlignment = xlCenter
         Worksheets("Data").Range("G27").Offset(0, j).HorizontalAlignment = xlCenter
         Worksheets("Data").Range("G27").Offset(j, 0).HorizontalAlignment = xlCenter
         Worksheets("Data").Range("G27").Offset(j, j).HorizontalAlignment = xlCenter
     Next j
   Next i
```

รูปที่ 12 (ต่อ) โค้ดของ User form ConfirmReset

```
            Worksheets("Run of Cycle").Range("C17") =Worksheets("Data").Range("K17").Value
        Worksheets("Run of Cycle").Range("C17").Font.Size = 14
        Worksheets("Run of Cycle").Range("C17").Font.Bold = True
        Worksheets("Run of Cycle").Range("C18") =Worksheets("Data").Range("K18").Value
        Worksheets("Run of Cycle").Range("C18").Font.Size = 14
        Worksheets("Run of Cycle").Range("C18").Font.Bold = True

Worksheets("Run of Cycle").Range("C20") = Worksheets("Data").Range("K21").Value
      Worksheets("Run of Cycle").Range("C20").Font.Size = 14
      Worksheets("Run of Cycle").Range("C20").Font.Bold = True

    Application.ScreenUpdating = True
End Sub
```

รูปที่ 12 (ต่อ) โค้ดของ User form ConfirmReset

4. โค้ดการทำงานของ Worksheets Run of Cycle

```
.....การประกาศตัวแปร
    Dim NumCustomer As Integer
    Dim NumVehicle As Integer
    Dim TypeSol As String
    Dim ImproveSol As String
.....การทำงานของปุ่ม Run
Public Sub Run2_Click()
    Dim TypeSol As String
    Dim ImproveSol As String
    Dim StartTime As Variant
    Dim EndTime As Variant
    Dim RunTime As Variant
StartTime = Timer    'เริ่มต้นการนับเวลาจากการ Run โปรแกรม

Call RandDistance 'สุ่มระยะทางจากลูกค้าจุดหนึ่ง ไปยัง ลูกค้าอีกจุดหนึ่ง
    Call InitialSolution ใส่หมายเลข 1,2,... และทำการสุ่ม InitialSolution

    Application.ScreenUpdating = False
    TypeSol = Worksheets("Run of Cycle").Range("F12").Value
    ImproveSol = Worksheets("Run of Cycle").Range("C22").Value
```

รูปที่ 13 การทำงานของ Worksheets Run of Cycle

```
'ถ้าผู้ใช้เลือกวิธีการค้นหาคำตอบแบบ Local Search 1
   If TypeSol = "Local Search 1" Then
      If ImproveSol = "แบบสลับ 1" Then
         Call LS1_Swap1
      ElseIf ImproveSol = "แบบสลับ 2" Then
         Call LS1_Swap2
      Else: ImproveSol = "แบบเลื่อน"
         Application.ScreenUpdating = False
         Call LS1_Slide
      End If
'ถ้าผู้ใช้เลือกวิธีการค้นหาคำตอบแบบ Local Search 2
   ElseIf TypeSol = "Local Search 2" Then
      If ImproveSol = "แบบสลับ 1" Then
         Call LS2_Swap1
      ElseIf ImproveSol = "แบบสลับ 2" Then
         Call LS2_Swap2
      Else: ImproveSol = "แบบเลื่อน"
         Call LS2_Slide
      End If
'ถ้าผู้ใช้เลือกวิธีการค้นหาคำตอบแบบ Iterated  Local Search
   Else: TypeSol = "Iterated Local Search"
      If ImproveSol = "แบบสลับ 1" Then
         Call ILS_Swap1
      ElseIf ImproveSol = "แบบสลับ 2" Then
         Call ILS_Swap2
      Else: ImproveSol = "แบบเลื่อน"
         Call ILS_Slide
      End If
End If
```

รูปที่ 13 (ต่อ) การทำงานของ Worksheets Run of Cycle

```
EndTime = Timer
RunTime = EndTime - StartTime  'สิ้นสุดการนับเวลาจากการ Run โปรแกรม
Worksheets("Run of Cycle").Cells(12, 3) = RunTime
Worksheets("Run of Cycle").Cells(13, 3) = (RunTime) / 60 'ให้แสดงเวลาเป็นนาที

Application.ScreenUpdating = True

End Sub

Private Sub Clear1_Click()
   'ล้างเซลล์ Sheet Run of Cycle
   Application.ScreenUpdating = False

   NumCustomer = Worksheets("Data").Range("K18").Value
   NumVehicle = Worksheets("Data").Range("K17").Value

     For i = 1 To 2 * NumCustomer + 4
      For j = 1 To NumCustomer + NumVehicle + 6
         Worksheets("Run of Cycle").Range("F13").Offset(i, j).ClearFormats
         Worksheets("Run of Cycle").Range("F13").Offset(i, j).ClearContents
      Next j
   Next i
     For i = 1 To 2 * NumCustomer + 4 Step 2
         Worksheets("Run of Cycle").Range("F14").Offset(i, 0).ClearFormats
         Worksheets("Run of Cycle").Range("F14").Offset(i, 0).ClearContents
     Next i
   Worksheets("Run of Cycle").Range("C12").ClearContents 'Timer
   Worksheets("Run of Cycle").Range("C13").ClearContents 'Timer
   Worksheets("Run of Cycle").Range("C17").ClearContents 'NumVehicle
   Worksheets("Run of Cycle").Range("C18").ClearContents 'NumCustomer
   Worksheets("Run of Cycle").Range("C20").ClearContents 'TotalCycle
   Worksheets("Run of Cycle").Range("C26").ClearContents 'Perturb
   Worksheets("Run of Cycle").Range("L12").ClearContents 'Perturb
   Worksheets("Run of Cycle").Range("Q12").ClearContents

     Application.ScreenUpdating = True
End Sub
```

รูปที่ 13 (ต่อ) การทำงานของ Worksheets Run of Cycle

```
Private Sub Back2_Click()
    Worksheets("Data").Activate
End Sub
```

รูปที่ 14 Back2

```
Private Sub Save2_Click()
        If MsgBox("คุณต้องการบันทึกผลลัพธ์ทั้งหมดใช่หรือไม่ ", vbOKCancel, "Save") =
   vbCancel Then Exit Sub
        ActivateWorkbook.Save
End Sub
```

รูปที่ 15 Save2

5. โค้ดการทำงานใน Module 1

```
Option Explicit
.....การประกาศตัวแปร
    Dim Distance() As Integer
    Dim LoSearch() As Integer
    Dim Solution() As Integer
    Dim NumVehicle As Integer
    Dim NumCustomer As Integer
    Dim NumberCustomer As Integer
    Dim r As Integer
    Dim rd As Single
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer
```

รูปที่ 16 การทำงานใน Module 1

```
Dim CurrentSol() As Integer
Dim NewSol() As Integer
Dim NewSolution() As Integer
Dim TempR As Integer
Dim Temp As Integer
Dim MaxCap As Integer
Dim TotalCycle As Integer
Dim TotalCap As Integer
Dim Num As Integer
Dim Cap As Integer
Dim X As Integer
Dim Y As Integer
Dim Z As Integer
Dim NowCust As Integer
Dim NextCust As Integer
Dim TotalDistance() As Single
Dim SumDistance As Single
Dim SumTD As Single
Dim BestTD As Integer
Dim BestNewTD As Single
Dim BestSolTD As Integer
Dim Best() As Integer '
Dim BestNewSol() As Integer
Dim BestSol() As Integer
Dim Row As Integer
Dim Column As Integer
```

รูปที่ 16 (ต่อ) การทำงานใน Module 1

5.1 โค้ดการสร้างตาราง

```
.....โค้ดการสร้างตาราง
Public Sub Table()
   With Selection.Borders(xlEdgeLeft)
            .LineStyle = xlContinuous
            .Weight = xlThin
            .ColorIndex = xlAutomatic
        End With
        With Selection.Borders(xlEdgeTop)
            .LineStyle = xlContinuous
            .Weight = xlThin
            .ColorIndex = xlAutomatic
        End With
        With Selection.Borders(xlEdgeBottom)
            .LineStyle = xlContinuous
            .Weight = xlThin
            .ColorIndex = xlAutomatic
        End With
        With Selection.Borders(xlEdgeRight)
            .LineStyle = xlContinuous
            .Weight = xlThin
            .ColorIndex = xlAutomatic
        End With
        With Selection.Borders(xlInsideVertical)
            .LineStyle = xlContinuous
            .Weight = xlThin
            .ColorIndex = xlAutomatic
        End With
        With Selection.Borders(xlInsideHorizontal)
            .LineStyle = xlContinuous
            .Weight = xlThin
            .ColorIndex = xlAutomatic
        End With
End Sub
```

รูปที่ 17 โค้ดการสร้างตาราง

5.2 โค้ดการสุ่มหาคำตอบเริ่มต้น

```
Public Sub InitialSolution()
  Dim LoSearch() As Integer
  Application.ScreenUpdating = False

   NumCustomer = Worksheets("Run of Cycle").Range("C18").Value
   NumVehicle = Worksheets("Run of Cycle").Range("C17").Value

ReDim LoSearch(1 To NumCustomer)

For i = 1 To NumCustomer
   Worksheets("Run of Cycle").Range("G14").Offset(0, i) = i
   Worksheets("Run of Cycle").Range("G14").Offset(0, i).Font.Size = 14
   Worksheets("Run of Cycle").Range("G14").Offset(0, i).Interior.ColorIndex = 44
   Worksheets("Run of Cycle").Range("G14").Offset(0, i).HorizontalAlignment = xlCenter
Next i

For i = 1 To NumCustomer
   LoSearch(i) = Worksheets("Run of Cycle").Range("G14").Offset(0, i).Value
Next i

'เริ่มทำการสุ่มคำตอบเริ่มต้น
For i = NumCustomer To 1 Step -1
      k = NumCustomer
      rd = Rnd()
      r = Int(rd * (k)) + 1
      If r = NumCustomer + 1 Then r = NumCustomer
      If TempR = r Then
            k = NumCustomer
            rd = Rnd()
            r = Int(rd * (k)) + 1
            If r = NumCustomer + 1 Then r = NumCustomer
       End If
```

รูปที่ 18 โค้ดการสุ่มหาคำตอบเริ่มต้น

```
        TempR = r
        Temp = LoSearch(r)
        LoSearch(r) = LoSearch(i)

 LoSearch(r) = LoSearch(i)
        LoSearch(i) = Temp
Next i

For i = 1 To NumCustomer
   Worksheets("Run of Cycle").Range("G16").Offset(0, i).Value = LoSearch(i)
   Worksheets("Run of Cycle").Range("G16").Offset(0, i).Font.Size = 14
   Worksheets("Run of Cycle").Range("G16").Offset(0, i).Interior.ColorIndex = 7
   Worksheets("Run of Cycle").Range("G16").Offset(0, i).HorizontalAlignment = xlCenter
Next i
```

รูปที่ 18 (ต่อ) โค้ดการสุ่มหาคำตอบเริ่มต้น

5.3 การคัดลอกชุดคำตอบเริ่มต้นเพื่อทำการปรับปรุงคำตอบ

```
Public Sub NewSolution1()
   NumCustomer = Worksheets("Run of Cycle").Range("C18").Value
   NumVehicle = Worksheets("Run of Cycle").Range("C17").Value

   ReDim CurrentSol(1 To NumCustomer)
   ReDim NewSolution(1 To 2 * NumCustomer, 1 To NumCustomer)

   For i = 1 To NumCustomer
      CurrentSol(i) = Worksheets("Run of Cycle").Range("G16").Offset(0, i).Value
   Next i
```

รูปที่ 19 การคัดลอกชุดคำตอบเริ่มต้นเพื่อทำการปรับปรุงคำตอบ

```
    For i = 1 To 2 * (NumCustomer) Step 2
        For j = 1 To NumCustomer
            NewSolution(i, j) = CurrentSol(j)
        Next j
    Next i
      For i = 1 To 2 * (NumCustomer) Step 2
        For j = 1 To NumCustomer
            Worksheets("Run of Cycle").Range("G17").Offset(i, j) = NewSolution(i, j)
        Next j
    Next i

End Sub
```

รูปที่ 19 (ต่อ) การคัดลอกชุดคำตอบเริ่มต้นเพื่อทำการปรับปรุงคำตอบ

5.4 การปรับปรุงคำตอบแบบสลับ 1

```
Public Sub Swap1()
    NumCustomer = Worksheets("Run of Cycle").Range("C18").Value
    NumVehicle = Worksheets("Run of Cycle").Range("C17").Value

  j = 1
For i = 1 To 2 * (NumCustomer) Step 2
    If j <= NumCustomer Then
        k = NumCustomer
        rd = Rnd()
        r = Int(rd * (k)) + 1
        If r = NumCustomer + 1 Then r = NumCustomer
            If j <> r Then
                Temp = Worksheets("Run of Cycle").Range("G17").Offset(i, j)
                Worksheets("Run of Cycle").Range("G17").Offset(i, j) = _
                Worksheets("Run of Cycle").Range("G17").Offset(i, r)
                Worksheets("Run of Cycle").Range("G17").Offset(i, r) = Temp
            End If
```

รูปที่ 20 การปรับปรุงคำตอบแบบสลับ 1

```
        If j = r Then
          Do While j = r
            k = NumCustomer
            rd = Rnd()
            r = Int(rd * (k)) + 1
            If r = NumCustomer + 1 Then r = NumCustomer
              Temp = Worksheets("Run of Cycle").Range("G17").Offset(i, j)
              Worksheets("Run of Cycle").Range("G17").Offset(i, j) = _
              Worksheets("Run of Cycle").Range("G17").Offset(i, r)
              Worksheets("Run of Cycle").Range("G17").Offset(i, r) = Temp
            Loop
        End If
        j = j + 1
      End If
    Next i

  End Sub
```

รูปที่ 20 (ต่อ) การปรับปรุงคำตอบแบบสลับ 1

5.5 การปรับปรุงคำตอบแบบสลับ 2

```
Public Sub Swap2()
    NumCustomer = Worksheets("Run of Cycle").Range("C18").Value
    NumVehicle = Worksheets("Run of Cycle").Range("C17").Value

    j = 1
    X = 1
For i = 1 To 2 * (NumCustomer) Step 2
    For X = 1 To 2
```

รูปที่ 21 การปรับปรุงคำตอบแบบสลับ 2

```
                k = NumCustomer
        rd = Rnd()
        r = Int(rd * (k)) + 1
        If r = NumCustomer + 1 Then r = NumCustomer
        If TempR = r Then

Do While TempR = r
                k = NumCustomer
                rd = Rnd()
                r = Int(rd * (k)) + 1
                If r = NumCustomer + 1 Then r = NumCustomer
            Loop
          End If
          TempR = r
        If j <> r Then
            Temp = Worksheets("Run of Cycle").Range("G17").Offset(i, j)
            Worksheets("Run of Cycle").Range("G17").Offset(i, j) = _
            Worksheets("Run of Cycle").Range("G17").Offset(i, r)
            Worksheets("Run of Cycle").Range("G17").Offset(i, r) = Temp
         End If
        If j = r Then
            Do While j = r
               k = NumCustomer
               rd = Rnd()
               r = Int(rd * (k)) + 1
               If r = NumCustomer + 1 Then r = NumCustomer
            Loop
```

รูปที่ 21 (ต่อ) การปรับปรุงคำตอบแบบสลับ 2

5.6 การปรับปรุงคำตอบแบบเลื่อน

```
Public Sub Slide()

   NumCustomer = Worksheets("Run of Cycle").Range("C18").Value
   NumVehicle = Worksheets("Run of Cycle").Range("C17").Value

   j = 1
For i = 1 To 2 * (NumCustomer) Step 2
     k = NumCustomer
     rd = Rnd()
     r = Int(rd * (k)) + 1
      If r = NumCustomer + 1 Then r = NumCustomer
     Temp = Worksheets("Run of Cycle").Range("G17").Offset(i, r)
        If r > j Then
           Do Until r = j
              Worksheets("Run of Cycle").Range("G17").Offset(i, r) = _
              Worksheets("Run of Cycle").Range("G17").Offset(i, r - 1)
              r = r - 1
           Loop
           Worksheets("Run of Cycle").Range("G17").Offset(i, j) = Temp
        End If
        If r < j Then
        Temp = Worksheets("Run of Cycle").Range("G17").Offset(i, r)
           Do Until r = j
              Worksheets("Run of Cycle").Range("G17").Offset(i, r) = _
              Worksheets("Run of Cycle").Range("G17").Offset(i, r + 1)
              r = r + 1
           Loop
        Worksheets("Run of Cycle").Range("G17").Offset(i, r) = Temp
     End If
     j = j + 1
Next i
End Sub
```

รูปที่ 22 การปรับปรุงคำตอบแบบเลื่อน

5.7 โค้ดการจัดลำดับของลูกค้าให้กับยานพาหนะโดยการพิจารณาความจุสินค้าที่ลูกค้าต้องการ

```
Public Sub CountCapacity()

NumCustomer = Worksheets("Run of Cycle").Range("C18").Value
NumVehicle = Worksheets("Run of Cycle").Range("C17").Value
MaxCap = Worksheets("Data").Range("K19").Value

For i = 0 To 2 * (NumCustomer) + 3 Step 2
   k = 0
   Temp = 0
   For j = 1 To NumCustomer
      If j < NumCustomer + 1 Then
         Do While Temp <= MaxCap
            Worksheets("Run of Cycle").Range("G15").Offset(i, j + k) = _
            Worksheets("Run of Cycle").Range("G14").Offset(i, j).Value
            Cap = Worksheets("Run of Cycle").Range("G15").Offset(i, j + k)
            TotalCap = Worksheets("Data").Range("C27").Offset(Cap, 0)
            TotalCap = Temp + TotalCap
            Temp = TotalCap
            j = j + 1
If j > NumCustomer Then
               If Temp > MaxCap Then
                  MaxCap = 1
               Else: MaxCap = 0
            End If
      End If
```

รูปที่ 23 โค้ดการจัดลำดับของลูกค้าให้กับยานพาหนะโดยการพิจารณาความจุสินค้าที่ลูกค้าต้องการ

```
    Loop
    If MaxCap = 1 Then
            Worksheets("Run of Cycle").Range("G15").Offset(i, j - 1 + k).ClearContents
            Worksheets("Run of Cycle").Range("G15").Offset(i, j + k) = _
            Worksheets("Run of Cycle").Range("G14").Offset(i, j - 1).Value
            MaxCap = 0
    End If
    If MaxCap <> 0 Then
          If Temp > MaxCap Then
             MaxCap = Worksheets("Data").Range("K19").Value
             TotalCap = 0
             Temp = 0
             Worksheets("Run of Cycle").Range("G15").Offset(i, j - 1 + k).ClearContents
             Worksheets("Run of Cycle").Range("G15").Offset(i, j + k) = _
             Worksheets("Run of Cycle").Range("G14").Offset(i, j - 1).Value
             Cap = Worksheets("Run of Cycle").Range("G15").Offset(i, j + k)
             TotalCap = Worksheets("Data").Range("C27").Offset(Cap, 0)
             TotalCap = Temp + TotalCap
             Temp = TotalCap
          End If
          Worksheets("Run of Cycle").Range("G15").Offset(i, j + k + 1) = _
          Worksheets("Run of Cycle").Range("G14").Offset(i, j).Value
          Cap = Worksheets("Run of Cycle").Range("G15").Offset(i, j + k + 1)
          TotalCap = Worksheets("Data").Range("C27").Offset(Cap, 0)
          TotalCap = Temp + TotalCap
          Temp = TotalCap
          k = k + 1
    End If
          MaxCap = Worksheets("Data").Range("K19").Value
    End If
    Next j
  Next i
End Sub
```

รูปที่ 23 (ต่อ) โค้ดการจัดลำดับของลูกค้าให้กับยานพาหนะโดยการพิจารณาความจุสินค้าที่ลูกค้าต้องการ

```
.....การใส่ 0 คั่น เมื่อยานพาหนะคันนั้นบรรจุสินค้าเต็มแล้ว
Public Sub MakeZero()
   NumCustomer = Worksheets("Run of Cycle").Range("C18").Value
   NumVehicle = Worksheets("Run of Cycle").Range("C17").Value

  For i = 1 To 2 * (NumCustomer) + 3 Step 2
   For j = 1 To NumVehicle + NumCustomer
      If Worksheets("Run of Cycle").Range("G14").Offset(i, j).Value = "" Then
         Worksheets("Run of Cycle").Range("G14").Offset(i, j) = "0"
      End If
   Next j
Next i
End Sub
```

รูปที่ 23 (ต่อ) โค้ดการจัดลำดับของลูกค้าให้กับยานพาหนะโดยการพิจารณาความจุสินค้าที่ลูกค้าต้องการ

5.8 โค้ดการคำนวณระยะทางในการเดินทางของยานพาหนะ

```
Public Sub SumTotalDistance()
Dim NowCust As Integer
Dim NextCust As Integer
Dim Distance As Integer
Dim SumDistance As Integer
Dim SumTD As Integer
Dim NewSol() As Integer
Dim BestSol() As Integer
Dim BestSolTD(1 To 3) As Integer
Dim TotalDistance() As Integer
Dim Z As Integer
Dim X As Integer
Dim Y As Integer
```

รูปที่ 24 โค้ดการคำนวณระยะทางในการเดินทางของยานพาหนะ

```
NumCustomer = Worksheets("Data").Range("K18").Value
NumVehicle = Worksheets("Data").Range("K17").Value

ReDim NewSol(1 To 2 * (NumCustomer), 1 To NumCustomer + NumVehicle)
ReDim TotalDistance(1 To 2 * NumCustomer)

ReDim BestSol(1 To 3, 1 To NumCustomer + NumVehicle)

For i = 1 To 3 Step 2
   For j = 1 To NumCustomer + NumVehicle
      BestSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j).Value
   Next j
Next i

For Z = 1 To 3 Step 2
   BestSolTD(Z) = Worksheets("Run of Cycle").Range("G14").Offset(Z, 0)
Next Z
'การหาระยะทางของคำตอบที่ดีที่สุดและคำตอบปัจจุบัน
For Z = 1 To 3 Step 2
      NowCust = 1
      For NextCust = 2 To NumCustomer + NumVehicle
         X = BestSol(Z, NowCust)
         Y = BestSol(Z, NextCust)
            SumDistance = Worksheets("Data").Range("G27").Offset(X + 1, Y + 1).Value
            SumTD = SumTD + SumDistance
         NowCust = NowCust + 1
      Next NextCust
      If NextCust = NumCustomer + NumVehicle + 1 Then
         X = BestSol(Z, NowCust)
         Y = BestSol(Z, 1)
         SumDistance = Worksheets("Data").Range("G27").Offset(X + 1, Y + 1).Value
         SumTD = SumTD + SumDistance
      End If
```

รูปที่ 24 (ต่อ) โค้ดการคำนวณระยะทางในการเดินทางของยานพาหนะ

```
        BestSolTD(Z) = SumTD
        SumTD = 0
Next Z

For Z = 1 To 3 Step 2
    Worksheets("Run of Cycle").Range("G14").Offset(Z, 0) = BestSolTD(Z)
Next Z

For i = 1 To 2 * (NumCustomer) Step 2
    For j = 1 To NumCustomer + NumVehicle
        NewSol(i, j) = Worksheets("Run of Cycle").Range("G18").Offset(i, j).Value
    Next j
Next i

For Z = 1 To (2 * (NumCustomer)) Step 2
    TotalDistance(Z) = Worksheets("Run of Cycle").Range("G18").Offset(Z, 0)
Next Z
'การหาระยะทางของคำตอบใหม่
For Z = 1 To (2 * (NumCustomer)) Step 2
        NowCust = 1
        SumTD = 0
        For NextCust = 2 To NumCustomer + NumVehicle
            X = NewSol(Z, NowCust)
            Y = NewSol(Z, NextCust)
                SumDistance = Worksheets("Data").Range("G27").Offset(X + 1, Y + 1).Value
                SumTD = SumTD + SumDistance
            NowCust = NowCust + 1
        Next NextCust
```

รูปที่ 24 (ต่อ) โค้ดการคำนวณระยะทางในการเดินทางของยานพาหนะ

```
        If NextCust = NumCustomer + NumVehicle + 1 Then
            X = NewSol(Z, NowCust)
            Y = NewSol(Z, 1)
            SumDistance = Worksheets("Data").Range("G27").Offset(X + 1, Y + 1).Value
            SumTD = SumTD + SumDistance
        End If

'ถ้าความต้องการของลูกค้าเกินจำนวนของยานพาหนะ
        If NextCust = NumCustomer + NumVehicle + 1 Then
            If Worksheets("Run of Cycle").Range("G18").Offset(Z, j).Value <> "" Then
                SumTD = SumTD * 1000
            End If
        End If
        TotalDistance(Z) = SumTD
Next Z


For Z = 1 To (2 * (NumCustomer)) Step 2  'ให้นำค่าที่คำนวณได้ไปแสดงบน Excel
    Worksheets("Run of Cycle").Range("G18").Offset(Z, 0) = TotalDistance(Z)
Next Z

End Sub
```

รูปที่ 24 (ต่อ) โค้ดการคำนวณระยะทางในการเดินทางของยานพาหนะ

5.9 การค้นหาคำตอบที่ดีที่สุดของ Local Search 1

```
Public Sub LS1()
Dim NumLS As Integer
Dim LS As Integer
Dim NumIt As Integer
Dim NewSolution() As Integer

    NumCustomer = Worksheets("Data").Range("K18").Value
    NumVehicle = Worksheets("Data").Range("K17").Value
    NumLS = Worksheets("Run of Cycle").Range("C20").Value

For LS = 1 To NumLS
    Call NewSolution1
    ReDim NewSol(1 To 2 * NumCustomer + 3, 1 To NumCustomer + NumVehicle)
    For i = 1 To 2 * (NumCustomer) + 3 Step 2
        For j = 1 To NumCustomer + NumVehicle
            NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j)
        Next j
    Next i

    For i = 1 To 2 * (NumCustomer) + 3 Step 2
        For j = 1 To NumCustomer + NumVehicle
            NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j).ClearContents
        Next j
    Next i

  Call CountCapacity
   Call MakeZero
   Call SumTotalDistance
```

รูปที่ 25 การค้นหาคำตอบที่ดีที่สุดของ Local Search 1

```
ReDim NewSol(1 To 2 * NumCustomer + 3, 1 To NumCustomer + NumVehicle)
 For i = 1 To 2 * (NumCustomer) + 3 Step 2
     For j = 1 To NumCustomer + NumVehicle
        NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j)
     Next j
  Next i

  For i = 1 To 2 * (NumCustomer) + 3 Step 2
      For j = 1 To NumCustomer + NumVehicle
         NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j).ClearContents
      Next j
   Next i

Call CountCapacity
 Call MakeZero
 Call SumTotalDistance

 ReDim Best(1 To NumCustomer)
 ReDim BestNewSol(1 To NumCustomer)
 ReDim BestSol(1 To 2 * NumCustomer, 1 To NumCustomer)
 ReDim NewSolution(1 To 2 * NumCustomer, 1 To NumCustomer)
 ReDim TotalDistance(1 To 2 * NumCustomer)

 For Z = 1 To (2 * (NumCustomer)) Step 2
    TotalDistance(Z) = Worksheets("Run of Cycle").Range("G18").Offset(Z, 0)
 Next Z
```

รูปที่ 25 (ต่อ) การค้นหาคำตอบที่ดีที่สุดของ Local Search 1

```
'การหาระยะทางที่สั้นที่สุด
    BestSolTD = TotalDistance(1)
      For j = 3 To (2 * (NumCustomer)) Step 2
        If TotalDistance(j) < BestSolTD Then
          BestSolTD = TotalDistance(j)
          Row = j
        End If
      Next j
      If BestSolTD = TotalDistance(1) Then
        Row = 1
      End If

    For i = 1 To 2 * NumCustomer Step 2
      For j = 1 To NumCustomer
        NewSolution(i, j) = Worksheets("Run of Cycle").Range("G17").Offset(i, j)
      Next j
    Next i

    For Column = 1 To NumCustomer
      BestNewSol(Column) = Worksheets("Run of Cycle").Range("G16").Offset(0, Column)
      Best(Column) = Worksheets("Run of Cycle").Range("G14").Offset(0, Column)
    Next Column

'ให้ระยะทางที่สั้นที่สุดของคำตอบใหม่มาแทนที่คำตอบปัจจุบัน
    BestNewTD = Worksheets("Run of Cycle").Range("G17").Value
    BestNewTD = BestSolTD
    Worksheets("Run of Cycle").Range("G17") = BestNewTD

    For Column = 1 To NumCustomer
      BestNewSol(Column) = NewSolution(Row, Column)
      Worksheets("Run of Cycle").Range("G16").Offset(0, Column) = BestNewSol(Column)
    Next Column
```

รูปที่ 25 (ต่อ) การค้นหาคำตอบที่ดีที่สุดของ Local Search 1

```
'เปรียบเทียบค่าคำตอบใหม่กับค่าคำตอบที่ดีที่สุด
   BestTD = Worksheets("Run of Cycle").Range("G15").Value
   If BestNewTD < BestTD Then
      BestTD = BestNewTD
      Worksheets("Run of Cycle").Range("G15") = BestTD
         For Column = 1 To NumCustomer
            Best(Column) = BestNewSol(Column)
            Worksheets("Run of Cycle").Range("G14").Offset(0, Column) = Best(Column)
         Next Column
         Worksheets("Run of Cycle").Range("Q12") = LS
   End If

   ReDim NewSol(1 To 2 * NumCustomer + 3, 1 To NumCustomer + NumVehicle)
   For i = 1 To 2 * (NumCustomer) + 3 Step 2
            For j = 1 To NumCustomer + NumVehicle
               NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j)
            Next j
         Next i
      For i = 1 To 2 * (NumCustomer) + 3 Step 2
         For j = 1 To NumCustomer + NumVehicle
            NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j).ClearContents
         Next j
      Next i

   Call CountCapacity
   Call MakeZero

Worksheets("Run of Cycle").Range("L12") = LS
   Worksheets("Run of Cycle").Range("L12").Font.Size = 14
Next LS

Application.ScreenUpdating = True
End Sub
```

รูปที่ 25 (ต่อ) การค้นหาคำตอบที่ดีที่สุดของ Local Search 1

5.10 การค้นหาคำตอบที่ดีที่สุดของ Local Search 2

```
Public Sub LS2 0
Dim NumLS As Integer
Dim LS As Integer
Dim NumIt As
Dim NewSolution() As Integer

   NumCustomer = Worksheets("Data").Range("K18").Value
   NumVehicle = Worksheets("Data").Range("K17").Value
   NumLS = Worksheets("Run of Cycle").Range("C20").Value
For LS = 1 To NumLS
   Call NewSolution1

 ReDim NewSol(1 To 2 * NumCustomer + 3, 1 To NumCustomer + NumVehicle)
   For i = 1 To 2 * (NumCustomer) + 3 Step 2
      For j = 1 To NumCustomer + NumVehicle
         NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j)
      Next j
   Next i

   For i = 1 To 2 * (NumCustomer) + 3 Step 2
      For j = 1 To NumCustomer + NumVehicle
         NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j).ClearContents
      Next j
   Next i
```

รูปที่ 26 การค้นหาคำตอบที่ดีที่สุดของ Local Search 2

```
Call CountCapacity

Call MakeZero
Call SumTotalDistance

ReDim Best(1 To NumCustomer)
ReDim BestNewSol(1 To NumCustomer)
ReDim BestSol(1 To 2 * NumCustomer, 1 To NumCustomer)
ReDim NewSolution(1 To 2 * NumCustomer, 1 To NumCustomer)
ReDim TotalDistance(1 To 2 * NumCustomer)

For Z = 1 To (2 * (NumCustomer)) Step 2
   TotalDistance(Z) = Worksheets("Run of Cycle").Range("G18").Offset(Z, 0)
Next Z
      BestSolTD = TotalDistance(1)
      For j = 3 To (2 * (NumCustomer)) Step 2
         If TotalDistance(j) < BestSolTD Then
            BestSolTD = TotalDistance(j)
            Row = j
         End If
      Next j
      If BestSolTD = TotalDistance(1) Then
         Row = 1
      End If

   For i = 1 To 2 * NumCustomer Step 2
      For j = 1 To NumCustomer
         NewSolution(i, j) = Worksheets("Run of Cycle").Range("G17").Offset(i, j)
      Next j
   Next i
```

รูปที่ 26 (ต่อ) การค้นหาคำตอบที่ดีที่สุดของ Local Search 2

```
For i = 1 To 2 * NumCustomer Step 2
    For j = 1 To NumCustomer
        NewSolution(i, j) = Worksheets("Run of Cycle").Range("G17").Offset(i, j)
    Next j
Next i

For Column = 1 To NumCustomer
    BestNewSol(Column) = Worksheets("Run of Cycle").Range("G16").Offset(0, Column)
    Best(Column) = Worksheets("Run of Cycle").Range("G14").Offset(0, Column)
Next Column

'ถ้าคำตอบใหม่ที่ค่าดีกว่าคำตอบปัจจุบัน ให้นำค่าคำตอบใหม่มาแทนที่คำตอบปัจจุบัน
    BestNewTD = Worksheets("Run of Cycle").Range("G17").Value
    If BestSolTD < BestNewTD Then
        BestNewTD = BestSolTD
        Worksheets("Run of Cycle").Range("G17") = BestNewTD
      For Column = 1 To NumCustomer
        BestNewSol(Column) = NewSolution(Row, Column)
        Worksheets("Run of Cycle").Range("G16").Offset(0, Column) = BestNewSol(Column)
    Next Column
    End If
```

รูปที่ 26 (ต่อ) การค้นหาคำตอบที่ดีที่สุดของ Local Search 2

```
'เปรียบเทียบค่าคำตอบใหม่กับค่าคำตอบที่ดีที่สุด
BestTD = Worksheets("Run of Cycle").Range("G15").Value
If BestNewTD < BestTD Then
    BestTD = BestNewTD
  Worksheets("Run of Cycle").Range("G15") = BestTD
    For Column = 1 To NumCustomer
      Best(Column) = BestNewSol(Column)
      Worksheets("Run of Cycle").Range("G14").Offset(0, Column) = Best(Column)
    Next Column
    Worksheets("Run of Cycle").Range("Q12") = LS
End If

ReDim NewSol(1 To 2 * NumCustomer + 3, 1 To NumCustomer + NumVehicle)
  For i = 1 To 2 * (NumCustomer) + 3 Step 2
    For j = 1 To NumCustomer + NumVehicle
      NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j)
    Next j
  Next i
  For i = 1 To 2 * (NumCustomer) + 3 Step 2
    For j = 1 To NumCustomer + NumVehicle
      NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j).ClearContents
    Next j
  Next i

  Call CountCapacity
  Call MakeZero

Worksheets("Run of Cycle").Range("L12") = LS
  Worksheets("Run of Cycle").Range("L12").Font.Size = 14
Next LS

End Sub
```

รูปที่ 26 (ต่อ) การค้นหาคำตอบที่ดีที่สุดของ Local Search 2

5.11 การค้นหาคำตอบที่ดีที่สุดของ Iterated Local Search

```
Public Sub ILS ()
Dim NumLS As Integer
Dim NumIt As Integer
Dim NewSolution() As Integer
Dim LoSearch() As Integer
Dim Perturbation As Integer
Dim Perturb As Integer
Dim ILS As Integer

   NumCustomer = Worksheets("Data").Range("K18").Value
   NumVehicle = Worksheets("Data").Range("K17").Value
   NumLS = Worksheets("Run of Cycle").Range("C20").Value

For ILS = 1 To NumLS
   Call NewSolution1

 ReDim NewSol(1 To 2 * NumCustomer + 3, 1 To NumCustomer + NumVehicle)
  For i = 1 To 2 * (NumCustomer) + 3 Step 2
         For j = 1 To NumCustomer + NumVehicle
            NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j)
         Next j
      Next i
    For i = 1 To 2 * (NumCustomer) + 3 Step 2
       For j = 1 To NumCustomer + NumVehicle
          NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j).ClearContents
       Next j
    Next i
```

รูปที่ 27 การค้นหาคำตอบที่ดีที่สุดของ Iterated Local Search

```
 Call CountCapacity
 Call MakeZero
 Call SumTotalDistance

 ReDim Best(1 To NumCustomer)
 ReDim BestNewSol(1 To NumCustomer)
 ReDim BestSol(1 To 2 * NumCustomer, 1 To NumCustomer)
 ReDim NewSolution(1 To 2 * NumCustomer, 1 To NumCustomer)
 ReDim TotalDistance(1 To 2 * NumCustomer)
 ReDim LoSearch(1 To NumCustomer)

 For Z = 1 To (2 * (NumCustomer)) Step 2
    TotalDistance(Z) = Worksheets("Run of Cycle").Range("G18").Offset(Z, 0)
 Next Z

 BestSolTD = TotalDistance(1)
    For j = 3 To (2 * (NumCustomer)) Step 2
       If TotalDistance(j) < BestSolTD Then
          BestSolTD = TotalDistance(j)
          Row = j
       End If
    Next j
    If BestSolTD = TotalDistance(1) Then
       Row = 1
    End If

 For i = 1 To 2 * NumCustomer Step 2
    For j = 1 To NumCustomer
       NewSolution(i, j) = Worksheets("Run of Cycle").Range("G17").Offset(i, j)
    Next j
 Next i
For Column = 1 To NumCustomer
    BestNewSol(Column) = Worksheets("Run of Cycle").Range("G16").Offset(0, Column)
    Best(Column) = Worksheets("Run of Cycle").Range("G14").Offset(0, Column)
 Next Column
```

รูปที่ 27 (ต่อ) การค้นหาคำตอบที่ดีที่สุดของ Iterated Local Search

```
'ให้ระยะทางของคำตอบใหม่ที่น้อยที่สุดมาแทนที่คำตอบปัจจุบัน
BestNewTD = Worksheets("Run of Cycle").Range("G17")
   If BestSolTD < BestNewTD Then
         BestNewTD = BestSolTD
         Worksheets("Run of Cycle").Range("G17") = BestNewTD
      For Column = 1 To NumCustomer
         BestNewSol(Column) = NewSolution(Row, Column)
         Worksheets("Run of Cycle").Range("G16").Offset(0, Column) =BestNewSol(Column)
      Next Column
   End If

'ถ้าค่าของคำตอบใหม่นั้นได้ค่าที่แยกว่าค่าคำตอบปัจจุบัน
 Perturbation = Worksheets("Run of Cycle").Range("C26").Value
 If BestSolTD >= BestNewTD Then
   Perturb = Perturb + 1
      If Perturb = Perturbation + 1 Then
         For i = 1 To NumCustomer
            LoSearch(i) = Worksheets("Run of Cycle").Range("G16").Offset(0, i).Value
         Next i
         For i = NumCustomer To 1 Step -1
            k = NumCustomer
            rd = Rnd()
            r = Int(rd * (k)) + 1
            If TempR = r Then r = r + 1
            If r = NumCustomer + 1 Then r = NumCustomer
            TempR = r
            Temp = LoSearch(r)
            LoSearch(r) = LoSearch(i)
            LoSearch(i) = Temp
         Next i
```

รูปที่ 27 (ต่อ) การค้นหาคำตอบที่ดีที่สุดของ Iterated Local Search

```
    For i = 1 To NumCustomer
        Worksheets("Run of Cycle").Range("G16").Offset(0, i).Value = LoSearch(i)
        Worksheets("Run of Cycle").Range("G16").Offset(0, i).Font.Size = 12
        Worksheets("Run of Cycle").Range("G16").Offset(0, i).HorizontalAlignment = xlCenter
    Next i
      Perturb = 0
      End If
End If

'ถ้าค่าของคำตอบปัจจุบันมีค่าดีกว่าคำตอบที่ดีที่สุด
BestNewTD = Worksheets("Run of Cycle").Range("G17").Value
    BestTD = Worksheets("Run of Cycle").Range("G15").Value
    If BestNewTD < BestTD Then
        BestTD = BestNewTD
        Worksheets("Run of Cycle").Range("G15") = BestTD
        For Column = 1 To NumCustomer
            Best(Column) = BestNewSol(Column)
            Worksheets("Run of Cycle").Range("G14").Offset(0, Column) = Best(Column)
        Next Column
        Worksheets("Run of Cycle").Range("Q12") = ILS
    End If

    For i = 1 To 2 * (NumCustomer) + 3 Step 2
            For j = 1 To NumCustomer + NumVehicle
                NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j)
            Next j
        Next i
        For i = 1 To 2 * (NumCustomer) + 3 Step 2
            For j = 1 To NumCustomer + NumVehicle
                NewSol(i, j) = Worksheets("Run of Cycle").Range("G14").Offset(i, j).ClearContents
            Next j
        Next i
    Call CountCapacity
    Call MakeZero

    Worksheets("Run of Cycle").Range("L12") = ILS
    Worksheets("Run of Cycle").Range("L12").Font.Size = 14
Next ILS

End Sub
```

รูปที่ 27 (ต่อ) การค้นหาคำตอบที่ดีที่สุดของ Iterated Local Search