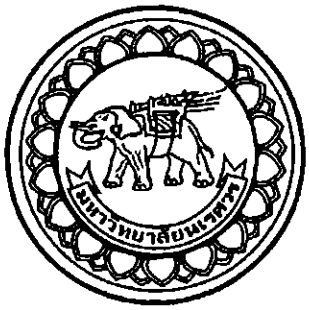


การหาคำตอบของปัญหาการเดินทางของพนักงานขายโดยวิธีเมตาฮิวริสติก

A METAHEURISTIC FOR TRAVELING SALESMAN PROBLEM

นายสายสัมพันธ์ ซุ้นเจริญ รหัส 52360645

นายอธิวัฒน์ กุลศิริ รหัส 52360768

ห้องสมุดคณะวิศวกรรมศาสตร์

วันที่รับ 24 ก.ค. 2556

เลขทะเบียน 16316499

เลขเรียกหนังสือ ฟร.

มหาวิทยาลัยนเรศวร ส663 ร 2555

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2555

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	การหาคำตอบของปัญหาการเดินทางของพนักงานขายโดยวิธีเมตาฮิวริสติก
ผู้ดำเนินโครงงาน	นายสายสัมพันธ์ ชุ้นเจริญ รหัส 52360645
	นายอธิวัฒน์ กุลศิริ รหัส 52360768
ที่ปรึกษาโครงงาน	ดร.ขวัญนิธิ คำเมือง
สาขา	วิศวกรรมอุตสาหการ
ภาควิชา	วิศวกรรมอุตสาหการ
ปีการศึกษา	2555

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ

.....ที่ปรึกษาโครงงาน

(ดร. ขวัญนิธิ คำเมือง)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. ภูพงษ์ พงษ์เจริญ)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ศรีสัจจา วิทยศักดิ์)

ชื่อหัวข้อโครงงาน	การหาคำตอบของปัญหาการเดินทางของพนักงานขายโดยวิธีเมตาฮิวริสติก
ผู้ดำเนินโครงงาน	นายสายสัมพันธ์ ชุ้นเจริญ รหัส 52360645
	นายอธิวัฒน์ กุลศิริ รหัส 52360768
ที่ปรึกษาโครงงาน	ดร.ขวัญนิธิ คำเมือง
สาขา	วิศวกรรมอุตสาหการ
ภาควิชา	วิศวกรรมอุตสาหการ
ปีการศึกษา	2555

บทคัดย่อ

โครงงานนี้ได้ศึกษาปัญหาการหาคำตอบของปัญหาการเดินทางของพนักงานขายโดยวิธีเมตาฮิวริสติก โดยมุ่งเน้นที่การจัดลำดับเส้นทางการเดินทางที่คำนึงถึงการหาเส้นทางที่เหมาะสมที่สุดสำหรับการเดินทาง ซึ่งจะทำให้ประหยัดเวลา และค่าใช้จ่ายในการเดินทาง สำหรับการหาคำตอบที่เหมาะสมที่สุดของปัญหาการเดินทาง ซึ่งในโครงงานนี้ จะใช้วิธีการที่จัดอยู่ในกลุ่มของเมตาฮิวริสติกที่มีวิธีการค้นหาคำตอบเฉพาะที่ (Local Search) และวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ (Iterated Local Search) มาเป็นพื้นฐานสำหรับการประยุกต์ใช้งาน

ในการดำเนินโครงงานจะเริ่มจากการศึกษาวิธีการเมตาฮิวริสติก แล้วศึกษาการสร้างตัวแทนคำตอบและการหาค่าคำตอบ จากนั้นจึงออกแบบเมตาฮิวริสติก ซึ่งในโครงงานนี้มีการออกแบบเมตาฮิวริสติก 3 แบบ แบบแรกกับแบบสอง เป็นการค้นหาคำตอบเฉพาะที่ ซึ่งทั้งสองแบบนี้จะมีส่วนที่แตกต่างกัน แบบสาม เป็นการหาคำตอบเฉพาะที่แบบวนรอบซ้ำ เมื่อออกแบบเมตาฮิวริสติกแล้ว จากนั้นนำข้อมูลที่ได้ออกแบบ มาเขียนโปรแกรม โดยการเขียนโปรแกรมการหาคำตอบของปัญหาการเดินทางของพนักงานขาย จะใช้ Visual Basic for Applications (VBA) บน Microsoft Excel

จากนั้นนำโปรแกรมการหาคำตอบของปัญหาการเดินทางของพนักงานที่ทำขึ้น มาทำการทดลองกับโจทย์ที่ได้หามาหรือทดลองกับโจทย์ที่ได้สุ่มขึ้น ซึ่งจากโจทย์ที่ทำการทดลองจะทำให้รู้ว่า มีระยะทางในการเดินทางเท่าไหร่ และจะได้เส้นทางการเดินทางที่ดีที่สุด

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี เพราะได้รับความช่วยเหลืออย่างดียิ่งจาก ดร.ขวัญนิธิ คำเมือง อาจารย์ที่ปรึกษาปริญญานิพนธ์ ซึ่งท่านได้ให้คำแนะนำ และข้อคิดเห็นต่างๆ ของการดำเนินงานตลอดมา ทำให้ปริญญานิพนธ์ฉบับนี้มีความสมบูรณ์และถูกต้อง

ขอขอบคุณอาจารย์คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ทุกท่าน และขอขอบคุณคุณพ่อ และคุณแม่ ที่สนับสนุนและให้กำลังใจเสมอมา

คณะผู้ดำเนินโครงงานวิศวกรรม
นายสายสัมพันธ์ ชุ้นเจริญ
นายอธิวัฒน์ กุลศิริ

มกราคม 2556

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์.....ก
บทคัดย่อภาษาไทย.....ข
กิตติกรรมประกาศ.....ค
สารบัญ.....ง
สารบัญตาราง.....ฉ
สารบัญรูป.....ช

บทที่ 1 บทนำ.....1
1.1 ความเป็นมาและความสำคัญของโครงงาน.....1
1.2 วัตถุประสงค์ของโครงงาน.....2
1.3 เกณฑ์ชี้วัดผลงาน.....2
1.4 เกณฑ์ชี้วัดผลสำเร็จ.....2
1.5 ขอบเขตของการทำโครงงาน.....2
1.6 สถานที่ในการดำเนินงานโครงงาน.....3
1.7 ระยะเวลาในการดำเนินโครงงาน.....3
1.8 ขั้นตอนและแผนการดำเนินโครงงาน (Gantt Chart).....3

บทที่ 2 หลักการและทฤษฎีเบื้องต้น.....4
2.1 ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem).....4
2.2 ปัญหาที่เกี่ยวเนื่องกับปัญหาการเดินทางของพนักงานขาย.....9
2.3 วิธีแก้ปัญหาในการหาค่าคำตอบที่ดีที่สุดของปัญหาการเดินทางของพนักงานขาย.....10
2.4 วิธีการแก้ปัญหาแบบฮิวริสติก (Heuristic Algorithm).....12
2.4.1 วิธีการปรับปรุงคำตอบโดยการสลับสองตำแหน่ง (2 – Opt).....12
2.4.2 วิธีการปรับปรุงคำตอบโดยการสลับสามตำแหน่ง (3 – Opt).....13
2.5 วิธีการเมตาฮิวริสติก (Metaheuristic Algorithms).....14
2.5.1 หลักการเบื้องต้นของเมตาฮิวริสติก.....14
2.5.2 การแบ่งวิธีการเมตาฮิวริสติก.....15
2.6 การค้นหาคำตอบเฉพาะที่ (Local Search).....15
2.7 การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ (Iterated Local Search : ILS).....16

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินโครงงาน....19

3.1 ศึกษาปัญหาการเดินทางของพนักงานขาย....20

3.2 ศึกษาวิธีการเมตาฮิวริสติก และการค้นหาคำตอบเฉพาะที่แบบวนซ้ำ....20

3.3 ออกแบบฮิวริสติก....20

3.4 ตรวจสอบความถูกต้องของการออกแบบเมตาฮิวริสติก....20

3.5 ทดลองแบบของเมตาฮิวริสติก....21

3.6 สรุปผลและนำเสนอผลงาน....21

บทที่ 4 ผลการดำเนินโครงงาน....22

4.1 การสร้างตัวแทนคำตอบและการหาค่าคำตอบ....22

4.2 ออกแบบเมตาฮิวริสติก....24

4.3 วิธีการปรับปรุงคำตอบแบบสลับ (SWAP) และแบบเลื่อน (SLIDE)....31

4.4 ความสามารถของโปรแกรมและวิธีการใช้งานอย่างย่อ....40

4.5 การออกแบบและผลการทดลองเพื่อเปรียบเทียบประสิทธิภาพของวิธีการหาคำตอบ....46

4.6 การวิเคราะห์ความแปรปรวน (ANOVA) ของผลลัพธ์โดยใช้โปรแกรม Minitab 16....47

บทที่ 5 บทสรุปและข้อเสนอแนะ....51

5.1 สรุปผลการดำเนินโครงงาน....51

5.2 ปัญหาที่พบระหว่างการดำเนินโครงงาน....51

5.3 แนวทางในการแก้ปัญหา....52

เอกสารอ้างอิง....53

ประวัติผู้ดำเนินโครงงาน....54

สารบัญตาราง

ตารางที่ หน้า

1.1 ขั้นตอนและแผนการดำเนินโครงงาน....3

2.1 ปัญหาการเดินทางของพนักงานขายแบบสมมาตร....7

2.2 ปัญหาการเดินทางของพนักงานขายแบบไม่สมมาตร....7

2.3 ระยะทางการเดินทางระหว่างเมืองแต่ละเมือง....9

4.1 สรุปผลการวิเคราะห์ความแปรปรวน....49

สารบัญรูป

รูปที่ หน้า

2.1 แผงวงจรอิเล็กทรอนิกส์....5
2.2 ปัญหาการเดินทางของพนักงานขายแบบสมมาตร....6
2.3 ปัญหาการเดินทางของพนักงานขายแบบไม่สมมาตร....7
2.4 รอบฮามิลโตเนียน ที่มีเส้นทางการเดินรอบใหญ่....8
2.5 รอบฮามิลโตเนียน ที่มีเส้นทางการเดินรอบย่อย....8
2.6 เส้นทางฮามิลโตเนียน....8
2.7 วิธีการสลับแบบ 2 – Opt....12
2.8 เส้นทางการเดินทางก่อนการปรับปรุง และหลังการปรับปรุงด้วยวิธี 2 – Opt....13
2.9 วิธีการสลับแบบ 3 – Opt....13
2.10 เส้นทางการเดินทางก่อนการปรับปรุง และหลังการปรับปรุงด้วยวิธี 3 – Opt....13
2.11 ลำดับขั้นตอนการทำงานโดยทั่วไปของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ....17
2.12 รหัสเทียมของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ....18
3.1 ขั้นตอนการดำเนินโครงงานของปัญหาการเดินทางของพนักงานขาย....19
4.1 แสดงตัวแทนคำตอบแบบกำหนดให้ n = 5....22
4.2 ตัวอย่างการสร้างตัวแทนคำตอบ....23
4.3 ตัวอย่างโจทย์การหาค่าคำตอบ....23
4.4 การคำนวณค่าคำตอบของระยะทางรวม....24
4.5 การทำงานของ Local Search Type 1....25
4.6 การทำงานของ Local Search Type 2....27
4.7 การทำงานของ Iterated Local Search....29
4.8 ตัวแทนคำตอบปัจจุบันแบบสลับ....31
4.9 การสลับโดยยึดตำแหน่ง 1 เป็นตัวหลัก....32
4.10 การสลับโดยยึดตำแหน่ง 2 เป็นตัวหลัก....33
4.11 การสลับโดยยึดตำแหน่ง 3 เป็นตัวหลัก....33
4.12 การสลับโดยยึดตำแหน่ง 4 เป็นตัวหลัก....35
4.13 การสลับโดยยึดตำแหน่ง 5 เป็นตัวหลัก....35
4.14 ตัวแทนคำตอบปัจจุบันแบบเลื่อน....36
4.15 การเลื่อนโดยยึดตำแหน่ง 1 เป็นตัวหลัก....36
4.16 การเลื่อนโดยยึดตำแหน่ง 2 เป็นตัวหลัก....37
4.17 การเลื่อนโดยยึดตำแหน่ง 3 เป็นตัวหลัก....38

สารบัญรูป (ต่อ)

รูปที่ หน้า

4.18 การเลื่อนโดยยึดตำแหน่ง 4 เป็นตัวหลัก....38
4.19 การเลื่อนโดยยึดตำแหน่ง 5 เป็นตัวหลัก....39
4.20 Exit Program....40
4.21 Save....41
4.22 New Table....41
4.23 Create Data....42
4.24 Input Data....42
4.25 Select Province....43
4.26 Number of Iteration....43
4.27 Choose Method....44
4.28 Number of Sub Iterated Local Search....44
4.29 Choose Type of Method....45
4.30 โปรแกรมจะทำการแก้ปัญหาการเดินทางของพนักงานขาย....45
4.31 การทดลองปัญหาขนาดเล็ก....46
4.32 การทดลองปัญหาขนาดกลาง....46
4.33 การทดลองปัญหาขนาดใหญ่....47
4.34 ตัวอย่างผลการวิเคราะห์รูปแบบของการหาคำตอบแบบสลับ และแบบเลื่อน....50

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem) เป็นปัญหารูปแบบหนึ่งในวิชาการวิจัยการดำเนินงาน (Operations Research) ไม่ว่าจะเป็นปัญหาการจัดเส้นทางการเดินทางเพื่อการค้าขายหรือส่งสินค้า ปัญหาการจัดเส้นทางการเดินทางเพื่อการท่องเที่ยว หรือแม้แต่การนำปัญหาการจัดเส้นทางนี้ ไปประยุกต์ใช้ในเครื่องมือ เครื่องจักร หรืออุปกรณ์อันทันสมัย เช่น การนำไปใช้ในเครื่องจักรสำหรับการกำหนดตำแหน่งของการทำงาน เพื่อให้ได้ประสิทธิภาพสูงสุด และปัญหาอื่นๆ อีกมากมาย ซึ่งปัญหาการเดินทางของพนักงานขายนี้จะเป็นปัญหาที่มีรูปแบบการเดินทางโดยเริ่มต้นจากจุดเริ่มต้นหนึ่ง ไปยังจุดอื่นๆ จนครบทุกจุด และสิ้นสุดการเดินทางที่จุดเดิม โดยมีเป้าหมายในการแก้ปัญหาในเรื่องของระยะทางในการเดินทาง และอาจรวมถึงค่าใช้จ่ายที่เกิดขึ้นสำหรับการเดินทาง ที่ต้องการการวางแผนเส้นทางการเดินทางที่จะช่วยลดเวลา และระยะทางในการเดินทาง การควบคุมค่าใช้จ่าย เพื่อการไปถึงจุดมุ่งหมายปลายทางอย่างรวดเร็ว และประหยัดที่สุดเท่าที่จะเป็นไปได้ โดยการกำหนดการวางแผนการจัดเส้นทางการเดินทางที่ดีได้นั้น จะต้องมีเส้นทางการเดินทางที่ไม่ซ้ำซ้อน วกไปวนมา ไม่ว่าจะเป็นการเดินทางไปหรือกลับ ซึ่งการวางแผนการเดินทางที่ดีจะช่วยลดการเกิดปัญหาข้างต้นได้ และยังเป็นการเพิ่มประสิทธิภาพในการเดินทางอีกด้วย ซึ่งการเดินทางในระยะทางใกล้ๆ จะเป็นเรื่องง่ายที่จะทำการคำนวณระยะทางในการเดินทาง แต่ถ้าหากการเดินทางเป็นการเดินทางระยะทางไกล และมีจุดมุ่งหมายปลายทางหลายๆ จุด การแก้ปัญหาด้วยวิธีการคำนวณจึงเป็นวิธีที่ยุ่งยากซับซ้อนที่จะหาระยะทางที่สั้นที่สุดในการเดินทาง เนื่องจากขอบเขตของปัญหาจะมีขนาดใหญ่ขึ้น มีการเชื่อมโยงหลายๆ จุดเข้าด้วยกัน

ปัญหาการเดินทางของพนักงานขาย จัดเป็นปัญหาแบบ NP – Complete และได้รับความสนใจจากนักวิจัยจำนวนมาก โดยรูปแบบของงานวิจัยส่วนใหญ่ เป็นการคิดค้น และนำเสนอวิธีการใหม่ๆ ที่ทำให้การแก้ปัญหาง่ายขึ้น และมีประสิทธิภาพมากขึ้น เช่น วิธีการกำหนดการพลวัต (Dynamic Programming) วิธีการแตกกิ่งและจำกัดขอบเขต (Branchand Bound) ซึ่งวิธีการเหล่านี้จัดเป็นวิธีการหาค่าคำตอบโดยอาศัยหลักการทางคณิตศาสตร์ ซึ่งพบว่ามีความยุ่งยากซับซ้อน ไม่เหมาะสมกับปัญหาที่มีขนาดใหญ่ เนื่องจากต้องใช้เวลาในการคำนวณเป็นเวลานาน และใช้หน่วยความจำมาก ทำให้การค้นหาคำตอบมีความเป็นไปได้ยาก และลำบากมากยิ่งขึ้น จึงได้มีการพัฒนาวิธีการหาค่าคำตอบโดยอาศัยหลักการประมาณค่าขึ้นมา สามารถแบ่งได้เป็นวิธีแบบคอนสครัคทีฟ และวิธีการหาคำตอบที่มีรูปแบบที่ไม่แน่นอน (Stochastic Search Algorithm) ซึ่งจะจัดอยู่ในกลุ่มของวิธีการแบบเมตาฮิวริสติก ที่จะทำการหาคำตอบที่ดีภายในพื้นที่ของคำตอบที่เป็นไปได้ และเป็นคำตอบที่ดีที่สุดในพื้นที่นั้น หรือคำตอบที่ใกล้เคียงคำตอบที่ดีที่สุด ภายในระยะเวลาอันสั้นจากการประมาณค่าคำตอบ

ซึ่งสามารถใช้ได้กับปัญหาที่หลายหลาย ถึงแม้ว่าการที่เมตาฮิวริสติกอาจมีระเบียบขั้นตอนมาตรฐานที่แน่นอน แต่เมื่อนำไปประยุกต์ใช้ในปัญหาที่แตกต่างกัน จะมีรายละเอียดของขั้นตอนย่อยที่แตกต่างกันออกไป เช่น วิธีระบบมด วิธีการเชิงพันธุกรรม วิธีการเลียนแบบการอบอ่อน วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ เป็นต้น

โครงงานนี้มุ่งเน้นที่การจัดลำดับเส้นทางการเดินทาง ที่คำนึงถึงการหาเส้นทางที่เหมาะสมที่สุดสำหรับการเดินทาง ซึ่งจะทำให้ประหยัดเวลา และค่าใช้จ่ายในการเดินทาง สำหรับการหาคำตอบที่เหมาะสมที่สุดของปัญหาการเดินทาง ซึ่งในโครงงานนี้ จะใช้วิธีการที่จัดอยู่ในกลุ่มของเมตาฮิวริสติกที่มีวิธีการค้นหาคำตอบเฉพาะที่ (Local Search) และวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ (Iterated Local Search) มาเป็นพื้นฐานสำหรับการประยุกต์ใช้งาน ที่จะสามารถหาค่าคำตอบที่มีความเหมาะสมจากหลายล้านคำตอบ

1.2 วัตถุประสงค์ของโครงงาน

ออกแบบและพัฒนาโปรแกรมการจัดเส้นทางการเดินทาง ของปัญหาการเดินทางของพนักงานขาย ด้วยวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

1.3 เกณฑ์ชี้วัดผลงาน (Output)

โปรแกรมช่วยในการจัดเส้นทาง สำหรับปัญหาการเดินทางของพนักงานขายโดยใช้วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

1.4 เกณฑ์ชี้วัดผลสำเร็จ (Outcome)

โปรแกรมสามารถค้นหาเส้นทาง โดยมีจุดประสงค์เพื่อหาระยะทางที่เหมาะสมที่สุด

1.5 ขอบเขตของการทำโครงงาน

1.5.1 โครงงานนี้เป็นการประยุกต์ใช้วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำในการแก้ปัญหาการจัดเส้นทางการเดินทางของพนักงานขาย ที่สามารถกำหนดจุดเริ่มต้น และจุดสิ้นสุดได้ตามความต้องการ โดยจะนำวิธีที่ได้ออกแบบและพัฒนา ไปทดลองประสิทธิภาพการหาคำตอบกับโจทย์ปัญหาที่ได้จำลองขึ้น

1.5.2 การใช้โปรแกรม Microsoft Excel ในการเขียนโปรแกรมด้วย Visual Basic for Applications ในการออกแบบโปรแกรมการจัดเส้นทางการเดินทางด้วยวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

1.6 สถานที่ในการดำเนินงานโครงงาน

ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

1.7 ระยะเวลาในการดำเนินโครงงาน

มิถุนายน 2555 – มกราคม 2556

1.8 ขั้นตอนและแผนการดำเนินโครงงาน (Gantt Chart)

ตารางที่ 1.1 แสดงขั้นตอนในการดำเนินโครงงาน

การดำเนินงาน	ช่วงเวลา							
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.
1.8.1 ศึกษาทฤษฎีเกี่ยวกับปัญหาการจัดเส้นทางการเดินทางของพนักงานขาย	←	—	→					
1.8.2 ศึกษาทฤษฎีและหลักการที่จำเป็นเกี่ยวกับวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ		←	—	→				
1.8.3 ออกแบบและพัฒนาโปรแกรมสำหรับแก้ปัญหาการจัดเส้นทางการเดินทางของพนักงานขาย ด้วยวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ			←	—	→			
1.8.4 การทดลองตามแผนการทดลอง					←	—	—	→
1.8.5 สรุปผลและข้อเสนอแนะ								↔

บทที่ 2
หลักการและทฤษฎีเบื้องต้น

2.1 ปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem)

ปัญหาการจัดเส้นทางการเดินทางของพนักงานขาย เป็นปัญหาด้านการจัดเส้นทางการเดินทาง โดยลักษณะของปัญหานั้นต้องการคำตอบที่เป็นรอบของทางการเดินทางของพนักงานขาย โดยที่พนักงานขายต้องเดินทางไปยังลูกค้าทุกคน คนละหนึ่งครั้งเท่านั้น และเส้นทางการเดินทางนั้นจะต้องย้อนกลับมายังที่จุดเริ่มต้น โดยมีวัตถุประสงค์ให้มีระยะทางหรือระยะเวลาในการเดินทางโดยรวมต่ำที่สุดโดยสามารถแสดงเป็นตัวแบบกำหนดการทางคณิตศาสตร์ ซึ่งจะกำหนดให้ ระยะทางหรือค่าใช้จ่ายในการเดินทางจากเมือง i ไปเมือง j เท่ากับ d_{ij} และ x_{ij} คือการเดินทางจากเมือง i ไปเมือง j มี V เป็นจุดยอด (Vertex) หรือจุดที่จะเดินทางไปมี E เป็นเส้นเชื่อมระหว่างจุดยอด (Edges) และมีสับเซต S = {i, j} ซึ่งมีตัวแบบทางคณิตศาสตร์ดังต่อไปนี้ (David J. Rader, 2010)

$$\text{Minimize } Z = \sum_{i}\sum_{j>i} d_{ij}x_{ij} \tag{2.1}$$

$$\text{Subject to} \quad \sum_{i<j} x_{ij} + \sum_{i>j} x_{ji} = 2,\ i \in V \tag{2.2}$$

$$\sum_{\substack{i,j\in S \\ (i,j)\in E}} x_{ij} \le |S|-1,\ S \subset V,\ 3 \le |S| \le |V|-3 \tag{2.3}$$

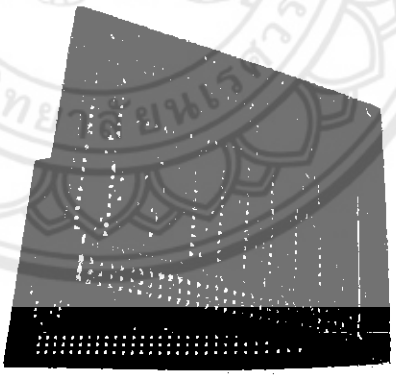
$$x_{ij} \in \{0,1\} \tag{2.4}$$

จากตัวแบบทางคณิตศาสตร์ จะเห็นได้ว่าสมการที่ 2.1 เป็นปัญหาการหาค่าคำตอบที่น้อยที่สุดของระยะทางหรือค่าใช้จ่ายในการเดินทางจากเมือง i ไปเมือง j และถ้าเมืองๆ ไหนที่มีการเดินทางไปจะถูกกำหนดให้มีค่าเท่ากับ 1 ส่วนเมืองที่เหลือซึ่งจะไม่มีการเดินทางไปให้มีค่าเท่ากับ 0 ทั้งหมด และระยะทางไปกลับมีค่าเท่ากันเสมอ โดยที่มีสมการที่ 2.2 บอกให้ทราบถึงว่าในเมืองหนึ่งๆ นั้น จะมีการเดินทางเข้ามาและออกไปจากเมืองๆ นั้นอย่างละครั้งเสมอ และมีสมการที่ 2.3 ไว้เพื่อป้องกันการเกิดรอบย่อย (Subtour) ของเส้นทางการเดินทาง กล่าวคือ เส้นทางของการเดินทางนั้น จะต้องเดินทางผ่านทุกจุดยอด 1 ครั้ง แล้วกลับมายังจุดเริ่มต้นเสมอ

โดยปัญหาจะมีเงื่อนไขว่า ในการเดินทางหนึ่งๆ ต้องไม่มีการใช้เส้นทางซ้ำซ้อน ไม่ว่าจะเป็นการเดินทางไปหรือกลับซึ่งคำตอบที่ดีที่สุดจะดูจากประสิทธิภาพของค่าคำตอบ จะวัดได้จากค่าของ

ฟังก์ชั่นเป้าหมาย (Objective Function) ซึ่งในปัญหาของการจัดเส้นทางการเดินทางของพนักงานขายนี้ จะเป็นปัญหาการหาค่าคำตอบที่น้อยที่สุด

ตัวอย่างของปัญหาการจัดเส้นทางการเดินทางของพนักงานขาย ซึ่งมีการนำเสนอขึ้นครั้งแรกโดย เซอร์วิลเลี่ยม ฮามิลตัน ในคริสศตวรรษที่ 18 ต่อมาในปี ค.ศ. 1930 คาร์ล เมนเกอร์ ได้ตีพิมพ์ปัญหานี้ในวารสาร และได้เป็นที่รู้จักในหมู่นักวิจัยทั่วไป ซึ่งในปี ค.ศ. 1934 เฮสเลอร์วิทย์นี่ ได้นำปัญหาการจัดเส้นทางการเดินทางของพนักงานขาย ไปบรรยายในมหาวิทยาลัย Princeton และในปี ค.ศ. 1937 งานวิจัยชิ้นแรกเกี่ยวกับปัญหานี้ ก็ถูกนำเสนอโดยเป็นปัญหาการจัดเส้นทางเดินรถรับส่งนักเรียนในรัฐนิวเจอร์ซี่ สหรัฐอเมริกา และจากการสำรวจทฤษฎีที่ได้มีผู้นำเทคนิคต่างๆ ในการแก้ปัญหามาประยุกต์ใช้อีกมากมาย เช่น การประยุกต์ใช้เป็นอุปกรณ์กำหนดแผนที่เส้นทางการเดินทางในรถยนต์ โดยใช้ระบบ Global Positioning System (GPS) ในการหาผลเฉลยของเส้นทางการเดินทางที่ได้กำหนดไว้ ให้เป็นเส้นทางการเดินทางที่ใกล้ที่สุด ที่จะสามารถบอกถึงลำดับของการเดินทาง จากการเปรียบเทียบระยะทางระหว่างจุดหนึ่งไปอีกจุดหนึ่ง และจุดอื่นๆ จากข้อมูลได้เก็บไว้ในระบบ หรือการใช้เป็นตัวนำทางในระบบของเครื่องจักรทางอุตสาหกรรมสมัยใหม่ ซึ่งมักจะมีการนำมาใช้ในระบบการผลิตของเครื่องจักรที่มีลักษณะชิ้นงานที่เหมือนๆ กัน ขนาดชิ้นงานเล็กๆ ยุ่งยากซับซ้อน และต้องทำหลายๆ ชิ้นติดต่อกันเป็นจำนวนมาก เช่น เครื่องติดชิพในแผงวงจรอิเล็กทรอนิกส์ ซึ่งจะเริ่มจากการติดชิพในตำแหน่งที่ใกล้ที่สุดในการหยิบจับชิพของแขนอัตโนมัติที่ใช้ในการหยิบจับชิพก่อน และค่อยๆ ติดในตำแหน่งที่ไกลออกไปจากตำแหน่งเดิมเรื่อยๆ จนครบทั่วทั้งแผงวงจร แสดงดังรูปที่ 2.1



รูปที่ 2.1 แผงวงจรอิเล็กทรอนิกส์

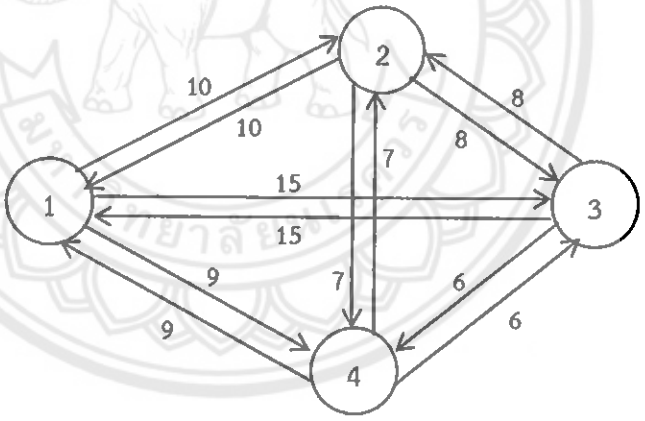
ที่มา : In Pursuitofthe Traveling Salesman, William J. Cook

นอกจากนี้ปัญหาการเดินทางของพนักงานขายนี้ ได้มีการพัฒนาต่อโดยได้แปลงปัญหาการเดินทางของพนักงานขายแบบทั่วไปเป็นปัญหาการจัดเส้นทางเดินของพนักงานขายแบบธรรมดาที่มีระยะทางไปกลับไม่เท่ากัน และปัญหาการจัดเส้นทางเดินของพนักงานขายหลายคนที่มีลักษณะแตกต่างกัน โดยปัญหาเหล่านี้ต่างก็มีวัตถุประสงค์เพื่อต้องการวางแผนเส้นทางการเดินทางที่ช่วยลด

เวลา และระยะทางในการเดินทาง การควบคุมค่าใช้จ่ายที่เกี่ยวข้องกับการเดินทาง และคาดหวังที่จะเพิ่มหรือพัฒนาประสิทธิภาพในการเดินทางให้ดีขึ้นเรื่อยๆ ในอนาคต

เป้าหมายหลักของปัญหาการจัดเส้นทางการเดินทางของพนักงานขาย คือ การหารอบของการเดินทางผ่านแต่ละจุด เพื่อให้ได้ต้นทุนต่ำที่สุด ถ้าต้นทุนของการเดินทางระหว่าง 2 ตำแหน่งไม่ขึ้นกับทิศทางในการเดินทาง โดยที่ระยะทาง หรือค่าใช้จ่ายในการไปกลับมีค่าเท่ากัน จะเรียกว่าปัญหาการเดินทางของพนักงานขายแบบสมมาตร (Symmetric Traveling Salesman Problem, STSP) แต่ถ้าไม่เท่ากัน แสดงว่าเป็นการเดินทางแบบมีทิศทาง จะเรียกว่าปัญหาการเดินทางของพนักงานขายแบบไม่สมมาตร (Asymmetric Traveling Salesman Problem, ATSP)

นิยามของปัญหาการจัดเส้นทางการเดินทางของพนักงานขาย จะมีลักษณะเป็นกราฟเส้นทาง ซึ่งกราฟจะมีอยู่ 2 รูปแบบ รูปแบบแรกเป็นแบบสมมาตร (Symmetric) และรูปแบบที่สองเป็นรูปแบบอสมมาตร (Asymmetric) ซึ่งการอธิบายในลักษณะของกราฟนั้นจะกำหนดให้กราฟ G มีระยะทางการเดินทางผ่านจุด i ไป j เท่ากับ d_{ij} โดยเราจะใช้คำว่า จุดยอด เป็นจุดต่างๆ ที่จะเดินทางไป และเส้นเชื่อม เป็นเส้นทางในการเดินทางไปสู่จุดยอดต่างๆ ซึ่งจะกำหนดให้ $G = (V, E)$ โดย $V = \{1, 2, ..., n\}$ โดยที่ n คือ จำนวนจุดยอดทั้งหมด และ $E = \{(i, j) : i, j \in V\}$ และ $d_{ij} = d_{ji}$ หรือ $d_{ij} \neq d_{ji}$ แสดงดังรูปที่ 2.2 (ธีรพล, 2547)



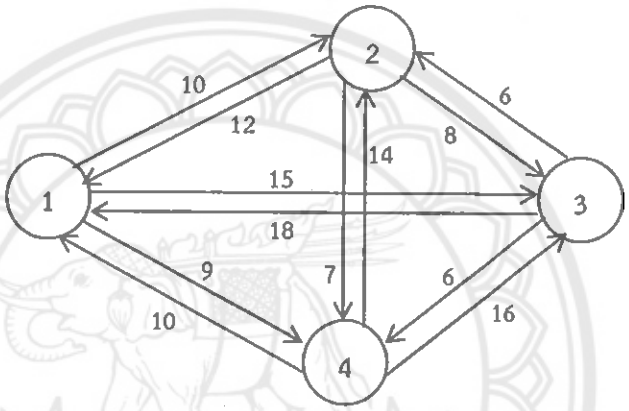
รูปที่ 2.2 ปัญหาการเดินทางของพนักงานขายแบบสมมาตร

จากรูปที่ 2.2 จะเห็นว่า การเดินทางจากเมือง 1 ไปเมืองที่ 2 มีค่าเท่ากับ 10 และการเดินทางจากเมือง 2 กลับมายังเมือง 1 มีค่าเท่ากับ 10 เช่นกัน และจะเป็นเช่นนี้ไปในทุกๆ กรณีของการเดินทางของระหว่างเมือง ภายในเมืองที่ 1 ถึงเมืองที่ 4 ลักษณะเช่นนี้ก็ก็ปัญหาการเดินทางของพนักงานขายแบบสมมาตร ซึ่งสามารถแสดงในรูปแบบของตารางได้ดังตารางที่ 2.1

ตารางที่ 2.1 ปัญหาการเดินทางของพนักงานขายแบบสมมาตร

	เมือง 1	เมือง 2	เมือง 3	เมือง 4
เมือง 1	-	10	15	9
เมือง 2	10	-	8	7
เมือง 3	15	8	-	6
เมือง 4	9	7	6	-

ในส่วนของปัญหาการเดินทางของพนักงานขายแบบไม่สมมาตร จะมีระยะทาง หรือค่าใช้จ่ายในการไปกลับมีค่าไม่เท่ากัน แสดงดังรูปที่ 2.3



รูปที่ 2.3 ปัญหาการเดินทางของพนักงานขายแบบไม่สมมาตร

จากรูปที่ 2.3 จะเห็นว่า การเดินทางจากเมือง 1 ไปเมือง 2 มีค่าเท่ากับ 10 และการเดินทางจากเมือง 2 กลับมายังเมือง 1 มีค่าเท่ากับ 12 ซึ่งมีระยะทาง หรือค่าใช้จ่ายในการไปกลับไม่เท่ากันในทุกๆ กรณีของการเดินทางของระหว่างเมือง ภายในเมืองที่ 1 ถึงเมืองที่ 4 และจะไม่มีการเดินทางไปเมืองๆ เดียวกันทุกกรณี ลักษณะเช่นนี้ก็จะเป็นปัญหาการเดินทางของพนักงานขายแบบไม่สมมาตรซึ่งสามารถแสดงในรูปแบบของตารางได้ดังตารางที่ 2.2

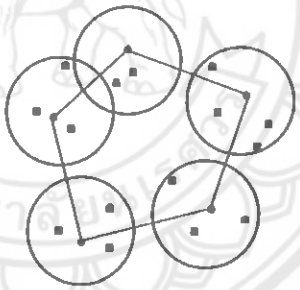
ตารางที่ 2.2 ปัญหาการเดินทางของพนักงานขายแบบไม่สมมาตร

	เมือง 1	เมือง 2	เมือง 3	เมือง 4
เมือง 1	-	10	15	9
เมือง 2	12	-	8	7
เมือง 3	18	6	-	6
เมือง 4	10	14	16	-

ในการมองรอบของเส้นทางการเดินทาง ถ้าเดินทางผ่านทุกจุดยอด 1 ครั้ง และกลับมาจุดเริ่มต้น เราจะเรียกว่ารอบฮามิลโตเนียน (Hamiltonian Cycle, HAMC) และถ้าหากรอบของเส้นทางการเดินทางนี้ ผ่านทุกจุดยอดทุกๆ จุดแล้วกลับมาที่จุดเริ่มต้น เส้นทางเดินที่ได้เราจะเรียกว่า รอบใหญ่ (Tour) แสดงดังรูปที่ 2.4

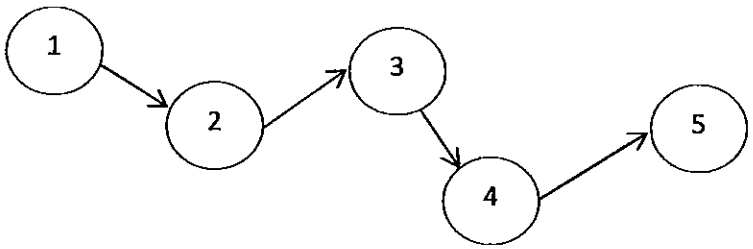
รูปที่ 2.4 รอบฮามิลโตเนียน ที่มีเส้นทางการเดินรอบใหญ่
ที่มา : In Pursuit of the Traveling Salesman, William J. Cook

แต่ถ้าการเดินทางไม่ผ่านครบทุกจุดยอดอย่างน้อย 1 ครั้ง แต่กลับมายังจุดเริ่มต้น เราจะเรียกว่า รอบย่อย แสดงดังรูปที่ 2.5



รูปที่ 2.5 รอบฮามิลโตเนียน ที่มีเส้นทางการเดินรอบย่อย
ที่มา : www.sciencedirect.com

ถ้าการเดินทางผ่านทุกจุดยอด 1 ครั้ง แต่ไม่กลับมาจุดเริ่มต้น เราจะเรียกกว่า เส้นทางฮามิลโตเนียน (Hamiltonian Path, HAMP) แสดงดังรูปที่ 2.6



รูปที่ 2.6 เส้นทางฮามิลโตเนียน

ตัวอย่างของปัญหาการเดินทางของพนักงานขาย

ถ้าพนักงานคนหนึ่งต้องเดินทางไปขายสินค้าที่เมืองต่างๆ 5 เมือง ซึ่งแต่ละเมืองมีระยะห่างระหว่างกันแสดงได้ดังตารางที่ 2.3 โดยให้จัดเส้นทางการเดินทางของพนักงานขายคนนี้ให้ได้เดินทางผ่านทุกเมือง และมีการเดินทางเข้าและออกเมืองละ 1 ครั้ง แล้วกลับมายังจุดเริ่มต้น

ตารางที่ 2.3 ระยะทางการเดินทางระหว่างเมืองแต่ละเมือง

i\j	1	2	3	4	5
1	0	10	17	20	21
2	10	0	21	14	18
3	16	98	0	19	20
4	17	14	20	0	17
5	18	18	19	16	0

จากตารางที่ 2.3 ระยะทางรวมของการเดินทางระหว่างเมือง สามารถหาได้โดยรวมระยะทางจากเมืองหนึ่งไปยังอีกเมืองหนึ่งไปจนครบเส้นทางการเดินทาง เช่น เส้นทางที่ 1 – 3 – 4 – 5 – 2 - 1 จะมีระยะทางระหว่างเมือง 1 - 3, 3 - 4, 4 - 5, 5 - 2 และ 2 - 1 ที่มีระยะทางเป็น 17, 19, 17, 18 และ 10 รวมเป็น 81 กิโลเมตร และระยะทางรวมของเส้นทางอื่นๆ สามารถคำนวณได้ด้วยวิธีเดียวกันนี้

ดังนั้น การแก้ปัญหาต้องทำการหาชุดคำตอบที่เป็นไปได้ทั้งหมด แล้วเลือกเส้นทางในการเดินทางที่มีระยะทางรวมน้อยที่สุดเป็นคำตอบ

2.2 ปัญหาที่เกี่ยวเนื่องกับปัญหาการเดินทางของพนักงานขาย

2.2.1 ปัญหาการเดินทางของพนักงานขายแบบ M พนักงาน

ปัญหาการเดินทางของพนักงานขายแบบ M พนักงาน (M – Traveling Salesman Problem, M – TSP) ซึ่งในปัญหาจะกำหนดให้มีพนักงานขายทั้งหมด M คน และจะกำหนดให้พนักงานขายแต่ละคนไม่มีข้อจำกัดเรื่องน้ำหนักบรรจุที่บรรจุได้สูงสุด โดยเราจะแปลงปัญหาการเดินทางของพนักงานขายจำนวน M พนักงาน ให้เป็นปัญหา TSP ก่อน แล้วค่อยทำการแก้ปัญหาดังกล่าว

2.2.2 ปัญหาการเดินทางของพนักงานขายแบบคอขวด

ปัญหาการเดินทางของพนักงานขายแบบคอขวด (Bottleneck Traveling Salesman Problem, BTSP) ถูกนำเสนอขึ้นครั้งแรกโดย Sphairo ในปี พ.ศ.2509 เป็นการหาคำตอบของปัญหา

การเดินทางของพนักงานขาย โดยให้ระยะทางจากเส้นทางที่อยู่ในคำตอบที่มีค่ามากที่สุดให้มีค่าน้อยที่สุด การแก้ปัญหาทำได้โดยการแก้ปัญหาหลายๆ ครั้งโดยแต่ละครั้งจะขจัดเส้นทางที่มีระยะทางมากที่สุดในคำตอบออก จนกระทั่งไม่พบคำตอบที่เป็นไปได้แล้ว เราจะถือว่าคำตอบสุดท้ายก่อนหน้าเป็นคำตอบที่ดีที่สุดของปัญหาการเดินทางของพนักงานขายแบบคอขวด

2.2.3 ปัญหาการเดินทางของพนักงานขายแบบเกาะกลุ่ม

ปัญหาการเดินทางของพนักงานขายแบบเกาะกลุ่ม (Clustered Traveling Salesman Problem, CTSP) เป็นปัญหาที่มีการแบ่งกลุ่ม หรือจุดยอดออกเป็นกลุ่มย่อยๆ โดยต้องเดินทางผ่านจุดยอดที่เป็นสมาชิกในแต่ละกลุ่มให้ครบก่อนเดินทางไปกลุ่มถัดไป

2.2.4 ปัญหาบุรุษไปรษณีย์ท้องถิ่น

ปัญหาบุรุษไปรษณีย์ท้องถิ่น (Rural Postman Problem, RPP) จากกราฟ (V, E) เราจะให้ $F \subseteq E$ โดยเราจะหาเส้นทางที่มีระยะทางรวมสั้นที่สุด โดยผ่านทุกเส้นทางในเซต F ซึ่งเราสามารถแปลงเป็นปัญหา TSP ได้ โดยเส้นทางในเซต F ที่เชื่อมระหว่างจุดยอด จะกำหนดไว้ในคำตอบของปัญหา TSP เสมอ และแก้ปัญหา TSP เพื่อหาเส้นทางเชื่อมระหว่างจุดยอดเข้าด้วยกัน

2.3 วิธีแก้ปัญหาในการหาค่าคำตอบที่ดีที่สุดของปัญหาการเดินทางของพนักงานขาย

ปัญหาการจัดเส้นทางการเดินทางของพนักงานขายจัดเป็น NP – Complete ซึ่งไม่มีวิธีการแก้ปัญหาให้ได้คำตอบที่ดีที่สุดภายในเวลาที่เหมาะสมและแน่นอนตายตัวได้ นั่นคือ เวลาที่ใช้ในการแก้ปัญหาขึ้นอยู่กับขนาด และความยากของปัญหานั้นๆ และเวลาจะเพิ่มขึ้นแบบเลขชี้กำลัง (Exponential) เมื่อชุดคำสั่งตัวอย่างของปัญหามีขนาดเพิ่มขึ้น

มีวิธีการหาค่าคำตอบที่ดีที่สุดสามารถแบ่งได้เป็น 2 วิธี คือ วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการทางคณิตศาสตร์ และวิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณค่า และเนื่องจากความล่าช้าในการค้นหาผลเฉลยที่ลงตัวของปัญหาของวิธีที่อาศัยหลักการทางคณิตศาสตร์ ดังนั้นวิธีการหาคำตอบโดยการประมาณค่าจึงได้ถูกนำมาใช้เพื่อค้นหาคำตอบให้ได้อย่างรวดเร็วยิ่งขึ้น ซึ่งที่นิยมนำมาใช้เป็นวิธีการสร้างคำตอบ จะจัดอยู่ในกลุ่มของวิธีการแบบเมตาฮิวริสติกที่มีลักษณะการทำงานเป็นการวนซ้ำของรอบการทำงานในการค้นหาค่าคำตอบ และจะหยุดเมื่อถึงเงื่อนไขที่กำหนดไว้ เพื่อหาหลักการในการเชื่อมจุดแต่ละจุดที่จะเดินทางไป ให้เข้ากันเป็นคำตอบที่ดีที่สุด ซึ่งจะขอกล่าวโดยสังเขป ดังนี้ (มาลินี และยุภาภัทร, 2554)

2.3.1 วิธีการหาคำตอบที่เหมาะสมที่สุดโดยหลักการทางคณิตศาสตร์ (Conventional Optimization Algorithms : COAs)

วิธีการนี้พบว่ามีความยุ่งยากซับซ้อน ไม่เหมาะสมกับปัญหาที่มีขนาดใหญ่ เนื่องจากต้องใช้เวลาในการคำนวณเป็นเวลานานและใช้หน่วยความจำมาก ซึ่งการแก้ปัญหาจะอยู่บนพื้นฐานของคณิตศาสตร์เป็นหลัก โดยมีขั้นตอน ตัวแปร และสมการที่ซับซ้อน แต่ถึงอย่างไรก็ตาม ความซับซ้อนของปัญหาในปัจจุบันที่มีมากมายมากขึ้น ทำให้วิธีการหาค่าคำตอบที่เหมาะสมที่สุดโดยวิธีการนี้ มีความเป็นไปได้ยาก และลำบากมากยิ่งขึ้น โดยเฉพาะในเรื่องของเวลาที่ใช้ในการประมวลผลค้นหาค่าคำตอบ จึงได้มีการพัฒนาวิธีการหาค่าคำตอบที่เหมาะสมที่สุดจากการประมาณค่าขึ้นมาใช้แทน ตัวอย่างของวิธีการหาคำตอบที่เหมาะสมที่สุดโดยหลักการทางคณิตศาสตร์ยกตัวอย่างเช่น

2.3.1.2 วิธีกำหนดการพลวัต (Dynamic Programming)

เป็นวิธีการที่ถูกพัฒนาขึ้นเพื่อใช้หาคำตอบที่เหมาะสมที่สุด ซึ่งพบว่าวิธีนี้จะเหมาะสำหรับปัญหาที่มีขนาดเล็ก วิธีการนี้ต้องใช้เวลาในการคำนวณมากและต้องใช้หน่วยความจำสูง ต่อมาได้ใช้วิธีการนี้เพื่อแก้ปัญหาการเดินทางของพนักงานขายที่มีเมืองอย่างน้อย 10 เมือง ผลที่ได้พบว่าเวลาที่ใช้ในการหาคำตอบเป็นการแจกแจงแบบเอ็กซ์โพเนนเชียลตามจำนวนเมืองที่เพิ่มขึ้น คือ ปัญหา 5เมือง ในเวลา 10 วินาที ปัญหา 10 เมือง ใช้เวลา 8 นาที สำหรับวิธีการนี้เมื่อมีจำนวนเมืองเพิ่มมากขึ้น เวลาที่ใช้จะคูณเวลาด้วยปัจจัยตัวหนึ่ง ซึ่งจะทำให้ความต้องการหน่วยความจำที่เพิ่มขึ้นด้วยความเร็วที่เหมือนกัน

2.3.1.3 วิธีการแตกกิ่งและจำกัดขอบเขต (Branch and Bound Method)

เป็นวิธีที่ใช้ในการหาคำตอบที่เหมาะสมที่สุด แต่เป็นวิธีที่ใช้เวลาและหน่วยความจำในการประมวลผลมาก ซึ่งในทางปฏิบัติจะนิยมใช้กับปัญหาขนาดเล็กเท่านั้น โดยได้มีการเสนออัลกอริทึมซึ่งได้ทำการแตกกิ่งเพื่อจำกัดทัวร์ย่อย แล้วใช้พื้นฐานการแก้ปัญหาลำดับของการมอบหมายงานซึ่งใช้วิธีการจำกัดขอบเขต จากนั้นใช้วิธีการแตกกิ่งและจำกัดขอบเขตโดยใช้เครื่องมือที่มีความแตกต่างกัน ซึ่งสามารถแก้ปัญหาได้มากสุด คือ 10 เมือง และไม่ทราบเวลาในการคำนวณ ดังนั้นการเปรียบเทียบประสิทธิผลจึงทำได้ยาก

2.3.2 วิธีการหาค่าคำตอบที่ดีที่สุดโดยอาศัยหลักการประมาณค่า (Approximation Optimization Algorithms : AOAs)

แบ่งเป็นวิธีการแบบคอนสตรัคทีฟ (Constructive Methods) ซึ่งเป็นทฤษฎี หรือหลักการที่ใช้เฉพาะเจาะจง (Specific Rule) กับปัญหา เพื่อใช้ในการสร้างผลเฉลย โดยจะค่อยๆ ทำการสร้างผลเฉลยขึ้นมาทีละส่วน จนกระทั่งผลเฉลยนั้นสมบูรณ์ เช่น วิธีการหาเส้นทางวิกฤต วิธีการวางแผนความต้องการวัสดุเป็นต้น และวิธีการหาคำตอบที่มีรูปแบบไม่แน่นอน (Stochastic Search Algorithms) จะจัดอยู่ในกลุ่มของวิธีการแบบเมตาฮิวริสติกซึ่งลักษณะการทำงานของวิธีการนี้ จะทำ

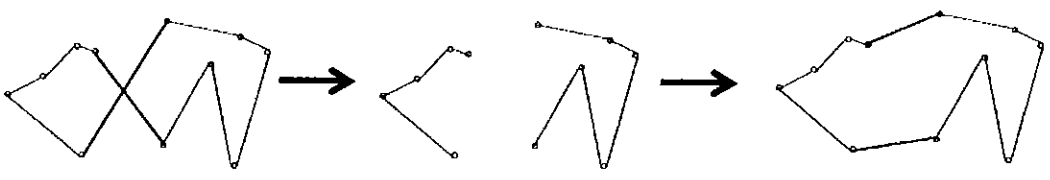
การวนซ้ำของรอบการทำงาน (Iteration) ในการค้นหาค่าคำตอบ และจะหยุดทำงานเมื่อถึงเงื่อนไขที่ได้กำหนดไว้ เช่น กำหนดให้มีการวนซ้ำกี่รอบ เป็นต้น

2.4 วิธีการแก้ปัญหาแบบฮิวริสติก (Heuristic Algorithm)

วิธีการแก้ปัญหาแบบฮิวริสติกนี้ เป็นวิธีการที่อาจจะได้คำตอบหรือไม่ได้คำตอบที่ดีที่สุด แต่จะใช้ระยะเวลาในการคำนวณที่สั้นกว่ามาก เมื่อเทียบกับแบบอาศัยหลักการทางคณิตศาสตร์ และมีลักษณะการทำงานแตกต่างกันมากมายหลายวิธี วิธีหนึ่งที่น่าสนใจได้ถูกนำเสนอโดย Linand Kernighan ซึ่งการนำมาใช้ในปัญหาการการเดินทางของพนักงานขายที่ได้นำเสนอนั้น สามารถแบ่งออกได้ 3 ขั้นตอน คือ เริ่มต้นจากการสร้างทัวร์ การปรับปรุงทัวร์ และการผสมกัน วิธีการแก้ปัญหานี้จะให้คำตอบใกล้เคียงคำตอบที่เหมาะสมที่สุด ใช้เวลาในการแก้ปัญหาและหน่วยความจำน้อย ซึ่งในงานวิจัยของ Lin and Kernighan (1973) ได้เสนอการนำวิธีการแก้ปัญหาการสุ่มอย่างมีเหตุผล ซึ่งมีแนวคิดดังนี้ที่จะนำวิธีการพื้นฐานมาแก้ปัญหาแบบผสมเพื่อให้ได้คำตอบที่เหมาะสมที่สุดโดยทำการปรับปรุงทีละขั้นจากการเลือกชุดเส้นทางที่เป็นไปได้ซึ่งขั้นตอนการหาคำตอบเป็นดังนี้ 1. สร้างเส้นทางที่เป็นไปได้ กำหนดให้เป็นชุดเส้นทาง T โดยทำการใส่เส้นทาง ซึ่งไม่มีในทัวร์ก่อนหน้านี้เข้าไปในทัวร์ใหม่ 2. พยายามทำการปรับปรุงเส้นทางเพื่อให้ได้เส้นทางใหม่เกิดขึ้น โดยทำการสลับเส้นทางในชุดเส้นทาง T โดยขั้นตอนการสลับกันได้ใช้หลักการแบบ 2 เส้นทาง (2 - Opt) และ 3 เส้นทาง (3 - Opt) 3. ถ้าการปรับปรุงพบว่าดีกว่าเดิม คือ ระยะทางหรือหน่วยวัดอื่นน้อยกว่าระยะทางหรือหน่วยวัดอื่นของเส้นทางชุด T ให้แทนที่ชุดเส้นทาง T ด้วยชุดเส้นทางใหม่ (T') แล้วกลับไปทำขั้นที่ 2 ต่อ 4 ถ้าไม่มีการปรับปรุงเส้นทาง แสดงว่าชุดเส้นทาง Tiเป็นคำตอบที่เหมาะสมที่สุดในเฉพาะที่ (Local Optimum) แล้วกระทำซ้ำในขั้น 1 จนเวลาการคำนวณหมดหรือจนได้คำตอบที่พอใจซึ่งมีวิธีการปรับปรุงคำตอบที่ได้ เพื่อให้ได้ผลเป็นที่น่าพอใจยิ่งขึ้นดังนี้ (ระพีพันธ์, 2554)

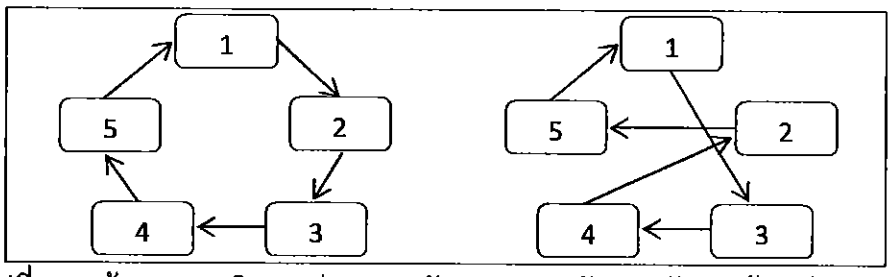
2.4.1 วิธีการปรับปรุงคำตอบโดยการสับสองตำแหน่ง (2 – Opt)

วิธีการสลับสองตำแหน่ง หรือเรียกว่า 2 – Opt (2 – Optimal) ซึ่งเป็นขั้นตอนวิธีการแก้ปัญหาขั้นพื้นฐานในการปรับปรุงเส้นทางในการเดินทาง โดยจะทำการเลือกตัดเส้นทางทีละ 2 คู่เส้นทางของรอบการเดินทางหนึ่งๆ และต่อเส้นทางขึ้นมาใหม่ครั้งละ 2 คู่เส้นทาง ที่มีระยะทางในการเดินทางสั้นลงกว่ารอบการเดินทางของเส้นทางเดิม โดยจะทำการตัด และต่อเส้นทางใหม่ไปเรื่อยๆ จนกว่าวิธีการสลับแบบ 2 – Opt จะไม่ให้คำตอบที่ดีขึ้นกว่าเดิมอีกแล้ว แสดงดังรูปที่ 2.7



รูปที่ 2.7 วิธีการสลับแบบ 2 – Opt

ตัวอย่างเช่น หากเส้นทางการเดินทางเป็น 1 – 2 – 3 – 4 – 5 - 1 การใช้วิธี 2 – Opt นี้ จะเลือกคู่ของเมืองมาสองคู่ เช่น ในที่นี้เลือกคู่ 1 - 2 และคู่ 4 - 5 ในการสลับ โดยหลักการคือ ห้ามคู่ 1 - 2 และ 4 - 5 เดินทางต่อกันอีก จะได้เส้นทางเดินใหม่ดังนี้ 3 – 4 – 2 – 5 - 1 หรือสามารถแสดงดังรูปที่ 2.8



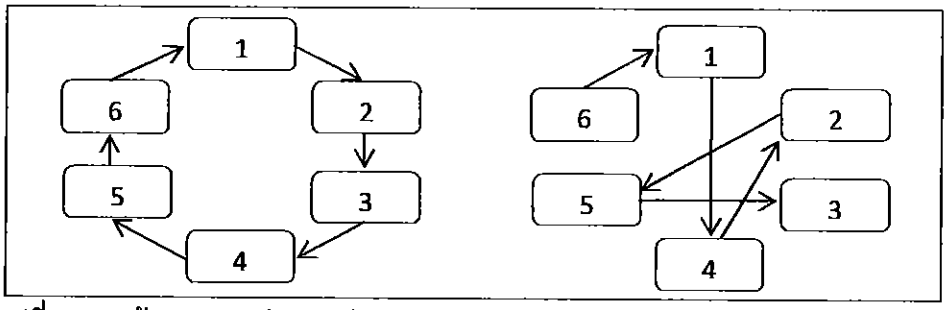
รูปที่ 2.8 เส้นทางการเดินทางก่อนการปรับปรุง และหลังการปรับปรุงด้วยวิธี 2 - Opt

2.4.2 วิธีการปรับปรุงคำตอบโดยการสลับสามตำแหน่ง (3 – Opt)

วิธีการสลับสามตำแหน่ง หรือเรียกว่า 3 - Opt (3 – Optimal) ซึ่งมีลักษณะของขั้นตอนวิธีการแก้ปัญหาเหมือนกับ 2 – Opt แต่จะทำการเลือกตัดเส้นทางทีละ 3 คู่เส้นทาง และต่อเส้นทางขึ้นมาใหม่ครั้งละ 3 คู่เส้นทาง โดยห้ามคู่ของทั้ง 3 เส้นทางที่เลือก เดินทางต่อกันอีก และทำการวนซ้ำการทำงานไปเรื่อยๆ จนกว่าจะได้รอบของเส้นทางที่ดีที่สุดในการเดินทางเมื่อเทียบกับการเดินทางในเส้นทางเดิม แสดงดังรูปที่ 2.9

รูปที่ 2.9 วิธีการสลับแบบ 3 – Opt

ตัวอย่างเช่น หากเส้นทางการเดินทางเป็น 1 – 2 – 3 – 4 – 5 – 6 - 1 ในการใช้วิธีการปรับปรุงแบบ 3 – Opt นี้จะเลือกคู่ของเมืองมาสามคู่ตามเงื่อนไข เช่น ในที่นี้จะเลือกคู่ 1 – 2, 3 - 4 และ 5 – 6 ในการสลับ โดยหลักการคือ ห้ามคู่ 1 - 2 และ 3 – 4 และ 5 – 6 เดินทางต่อกันอีก จะได้เส้นทางเดินใหม่ดังนี้ 6 – 1 – 4 – 2 – 5 – 3 หรือสามารถแสดงดังรูปที่ 2.10



รูปที่ 2.10 เส้นทางการเดินทางก่อนการปรับปรุง และหลังการปรับปรุงด้วยวิธี 3 - Opt

2.5 วิธีการเมตาฮิวริสติก (Metaheuristic Algorithms)

วิธีการฮิวริสติก เป็นระเบียบวิธีแบบอิสระที่สามารถสร้างกรรมวิธีหรือขั้นตอนใดๆ ก็ได้ เปรียบเสมือนการออกแบบเสื้อผ้า เมื่อออกแบบมาแล้วต้องสวยงาม ใช้ได้จริง และสวมใส่ง่าย เช่นเดียวกับการออกแบบวิธีการทางฮิวริสติกที่ต้องออกแบบให้ใช้ง่าย มีประสิทธิภาพ ใช้งานได้จริง

เมตาฮิวริสติก หมายถึง ชุดของลำดับขั้นการแก้ปัญหาแบบฮิวริสติกชนิดหนึ่งที่สามารถนำหลักการเดียวกันไปใช้แก้ปัญหาได้หลากหลายปัญหา ซึ่งในปัจจุบัน วิธีการออกแบบฮิวริสติกโดยอาศัยหลักการทางเมตาฮิวริสติกนี้ ได้รับความนิยมเป็นอย่างสูง เนื่องจากคำตอบที่ได้จากวิธีการนี้ให้ผลที่ดี แก้ปัญหาได้รวดเร็ว และใช้งานง่าย (ระพีพันธ์, 2554)

2.5.1 หลักการเบื้องต้นของเมตาฮิวริสติก

2.5.1.1 เมตาฮิวริสติกมีระเบียบวิธีในการค้นหาคำตอบที่ดีภายในพื้นที่ของคำตอบที่เป็นไปได้ (Feasible Region)

2.5.1.2 เมตาฮิวริสติกมีจุดประสงค์เพื่อหาคำตอบที่ดีที่สุด หรือคำตอบที่ใกล้เคียงคำตอบที่ดีที่สุดภายในระยะเวลาอันสั้น

2.5.1.3 วิธีการทางเมตาฮิวริสติกจะมีทั้งแบบง่าย ไม่ซับซ้อน เช่น การปรับปรุงคำตอบเฉพาะที่ (Local Search) หรือแบบที่ยุ่งยากซับซ้อนมากกว่า เช่น วิธีระบบมด (Ant Colony System) วิธีการเชิงพันธุกรรม (Genetic Algorithm) วิธีการค้นหาต้องห้าม (Tabu Search) วิธีการเลียนแบบการอบอ่อน (Simulated Annealing) เป็นต้น

2.5.1.4 เมตาฮิวริสติกเป็นขั้นตอนการประมาณคำตอบ

2.5.1.5 เมตาฮิวริสติกอาจจะเกิดจากการรวมหลากหลายเทคนิค เพื่อค้นหาคำตอบที่ดีที่สุดภายในพื้นที่คำตอบที่เป็นไปได้

2.5.1.6 เมตาฮิวริสติกมีระเบียบขั้นตอนมาตรฐานที่แน่นอน แม้ว่าเมื่อนำไปประยุกต์ใช้ในปัญหาที่แตกต่างกันจะมีรายละเอียดของขั้นตอนย่อยที่แตกต่างกัน แต่ฮิวริสติกสำหรับปัญหาแต่ละปัญหาต้องดำเนินการตามขั้นตอนหลักของเมตาฮิวริสติกดั้งเดิม

2.5.1.7 เมตาฮิวริสติกต้องสามารถใช้ได้กับปัญหาที่หลากหลาย

2.5.1.8 เมตาฮิวริสติกอาจจะมีลักษณะเป็นคำบรรยายโดยย่อก็ได้ หรือไม่จำเป็นต้องมีหลักการทางคณิตศาสตร์

2.5.1.9 ปัจจุบัน เมตาฮิวริสติกใช้ความจำชั่วคราวมากขึ้นในการจำคำตอบเดิม เพื่อค้นหาคำตอบที่ไม่ซ้ำเดิมหรือแตกต่างไปจากเดิม เช่น วิธีการค้นหาต้องห้าม วิธีระบบมด

ในปัจจุบัน วิธีการทางด้านเมตาฮิวริสติกมีอยู่หลากหลาย ซึ่งแต่ละวิธีการจะมีจุดดีและจุดด้อยที่แตกต่างกัน บางวิธีการให้ผลดีแต่ใช้เวลาในการคำนวณนาน บางวิธีการรวดเร็วแต่ให้ผลที่แย่กว่าวิธีอื่นๆ

2.5.2 การแบ่งวิธีการเมตาฮิวริสติก

2.5.2.1 แบบที่เกิดจากแรงบันดาลใจจากธรรมชาติ หรือแบบไม่ได้เกิดจากแรงบันดาลใจจากธรรมชาติ เช่น วิธีระบบมด วิธีการเชิงพันธุกรรม วิธีการเลียนแบบการอบอ่อน ส่วนวิธีการที่ไม่ได้เลียนแบบธรรมชาติ เช่น วิธีการค้นหาต้องห้าม การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ เป็นต้น

2.5.1.2 แบบใช้ประชากรหรือแบบไม่ใช้ประชากร โดยแบบใช้ประชากร คือ ในหนึ่งรอบของการคำนวณจะมีคำตอบมากกว่าหนึ่งคำตอบให้เลือก ส่วนแบบไม่มีประชากร คือ ในหนึ่งรอบของการคำนวณจะมีคำตอบเพียงคำตอบเดียว วิธีแบบใช้ประชากร เช่น วิธีระบบมด วิธีการเชิงพันธุกรรม ส่วนวิธีแบบไม่ใช้ประชากร เช่น วิธีการเลียนแบบการอบอ่อน วิธีการค้นหาต้องห้าม การค้นหาคำตอบเฉพาะที่แบบวนซ้ำ

2.5.1.3 แบบสมการเป้าหมายคงที่ หรือไม่คงที่ โดยในหนึ่งรอบของการคำนวณ อาจจะมีการเปลี่ยนแปลงสมการเป้าหมาย เพื่อให้ได้คำตอบใหม่ๆ เกิดขึ้น หรือไม่มีการเปลี่ยนแปลงสมการเป้าหมาย ในกรณีที่มีการเปลี่ยนสมการเป้าหมาย เช่น การค้นหาคำตอบเฉพาะที่แบบมีการชี้นำ (Guide Local Search) กรณีที่ไม่มีการเปลี่ยนสมการเป้าหมาย เช่น วิธีระบบมด วิธีการเชิงพันธุกรรม วิธีการเลียนแบบการอบอ่อน วิธีการค้นหาแบบต้องห้าม วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

2.5.1.4 แบบเปลี่ยนวิธีการหาคำตอบใกล้เคียงคำตอบปัจจุบัน (Neighborhood) คงที่และไม่คงที่ กรณีที่วิธีการหาคำตอบใกล้เคียงคงที่ ได้แก่ วิธีระบบมด วิธีการเชิงพันธุกรรม วิธีการเลียนแบบการอบอ่อน วิธีการค้นหาต้องห้าม วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ ส่วนกรณีที่มีการเปลี่ยนวิธีการหาคำตอบที่ใกล้เคียง ได้แก่ วิธีการค้นหาจากคำตอบใกล้เคียงแบบมีเงื่อนไข

2.5.1.5 แบบมีและไม่มีหน่วยความจำ จะจำว่ามีคำตอบใดบ้างที่ผ่านมาแล้วเพื่อเป็นข้อมูลในการหาคำตอบถึงไป ซึ่งในกรณีที่ใช้หน่วยความจำที่ปรากฏชัด ได้แก่ วิธีระบบมด วิธีการค้าหาต้องห้าม ส่วนวิธีที่ไม่ใช้หน่วยความจำ เช่น วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

สำหรับในงานวิจัยนี้จะใช้แบบเปลี่ยนวิธีการหาคำตอบใกล้เคียงคำตอบปัจจุบัน คือ วิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ ซึ่งจะมีการรบกวนคำตอบที่ได้ เพื่อให้ออกจากพื้นที่คำตอบเดิม และวนซ้ำการทำงานไปเรื่อยๆ ในการหาคำตอบใหม่ๆ ที่จะเกิดขึ้น ซึ่งจะได้อธิบายในรายละเอียดในหัวข้อถัดไป

2.6 การค้นหาคำตอบเฉพาะที่ (Local Search)

เมตาฮิวริสติกเป็นลำดับวิธีในการหาคำตอบที่ดีภายในระยะเวลาจำกัด เป็นวิธีการหาคำตอบจากคำตอบที่เป็นไปได้ (Feasible Solution) ในพื้นที่คำตอบที่เป็นไปได้จะมีคำตอบหนึ่งที่เป็นคำตอบที่มีค่าที่ดีที่สุด เรียกว่า คำตอบที่ดีที่สุดในพื้นที่คำตอบที่เป็นไปได้ (Global Optimum) นอกจากนี้หากพื้นที่คำตอบที่มีคำตอบที่เป็นไปได้มีเนื้อที่ใหญ่มากหรือมีจำนวนคำตอบที่เป็นไปได้มาก ก็อาจมีการแยกย่อยพื้นที่ออกเป็นพื้นที่เล็กๆ และคำตอบที่ดีที่สุดในแต่ละพื้นที่ที่ถูกแบ่งนั้น จะเรียกว่า คำตอบที่

ดีที่สุดในพื้นที่ย่อย (Local Optimum) ซึ่งวิธีการค้นหาคำตอบที่ดีที่สุดในพื้นที่ที่ถูกแบ่งย่อยนั้นจะเรียกว่า การค้นหาคำตอบเฉพาะที่ (Local Search) (ระพีพันธ์, 2554)

ซึ่งฮิวริสติกนี้อาจจะได้คำตอบที่ดีขึ้นที่หากมีการเพิ่มการค้นหาคำตอบเฉพาะที่เข้าไปในฮิวริสติก ซึ่งจะได้วิธีการที่สมบูรณ์มากขึ้น การค้นหาคำตอบเฉพาะที่สำหรับปัญหา TSP มีอยู่หลายวิธีซึ่งมีความยุ่งยากซับซ้อนแตกต่างกัน เช่น 2 - Opt, 3 - Opt, Swap เป็นต้น

2.7 การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ (Iterated Local Search : ILS)

การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำพัฒนามาจากฮิวริสติกแบบค้นหาในพื้นที่บางส่วนในพื้นที่คำตอบที่เป็นไปได้ขั้นพื้นฐาน (Basic Local Search, BLS) โดยที่แนวคิดของการค้นหาในพื้นที่บางส่วนในพื้นที่คำตอบที่เป็นไปได้ขั้นพื้นฐานคือการหาจุดที่ดีที่สุดในพื้นที่หนึ่งที่จำกัดในพื้นที่ที่เป็นไปได้ ซึ่งการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำได้นำเอาข้อดีของการค้นหาในพื้นที่บางส่วนในพื้นที่คำตอบที่เป็นไปได้ขั้นพื้นฐานมาใช้หาคำตอบที่ดีที่สุดในแต่ละพื้นที่ หลังจากนั้นจะพัฒนาคำตอบที่มีอยู่ เพื่อให้ออกจากพื้นที่เดิม (Escape) แล้วค้นหาคำตอบที่ดีที่สุดในพื้นที่ใหม่ ตัวอย่างเช่น วิธีการหาอาหารของกบ สมมุติให้กบกระโดดไปหากินแมลงที่จุดๆหนึ่ง หลังจากกินแมลงจุดนั้นหมด กบก็จะกระโดดไปหาพื้นที่ใกล้ๆ และกินแมลงรอบๆ บริเวณใหม่นั้นจนหมด และกระโดดไปยังพื้นที่ใกล้ๆต่อไปเรื่อยๆ การที่กบกระโดดไปและหาอาหารกินรอบๆ ตัวนั้นก็คือการค้นหาคำตอบเฉพาะที่รอบๆบริเวณคำตอบ การกระโดดไปหาพื้นที่ใหม่ใกล้ๆ พื้นที่เดิมคือการพัฒนาคำตอบที่มีอยู่ เช่นเดียวกับกลักการของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำที่มีการหาคำตอบบริเวณรอบๆ ซึ่งก็คือการหาคำตอบเฉพาะที่ และพัฒนาคำตอบเดิมเพื่อให้ได้พื้นที่ใหม่ในการค้นหาคำตอบเฉพาะที่ จากนั้นทำซ้ำไปเรื่อยๆ จนกระทั่งผู้ออกแบบฮิวริสติกพอใจ จึงหยุดการพัฒนาคำตอบและการค้นหาคำตอบเฉพาะที่

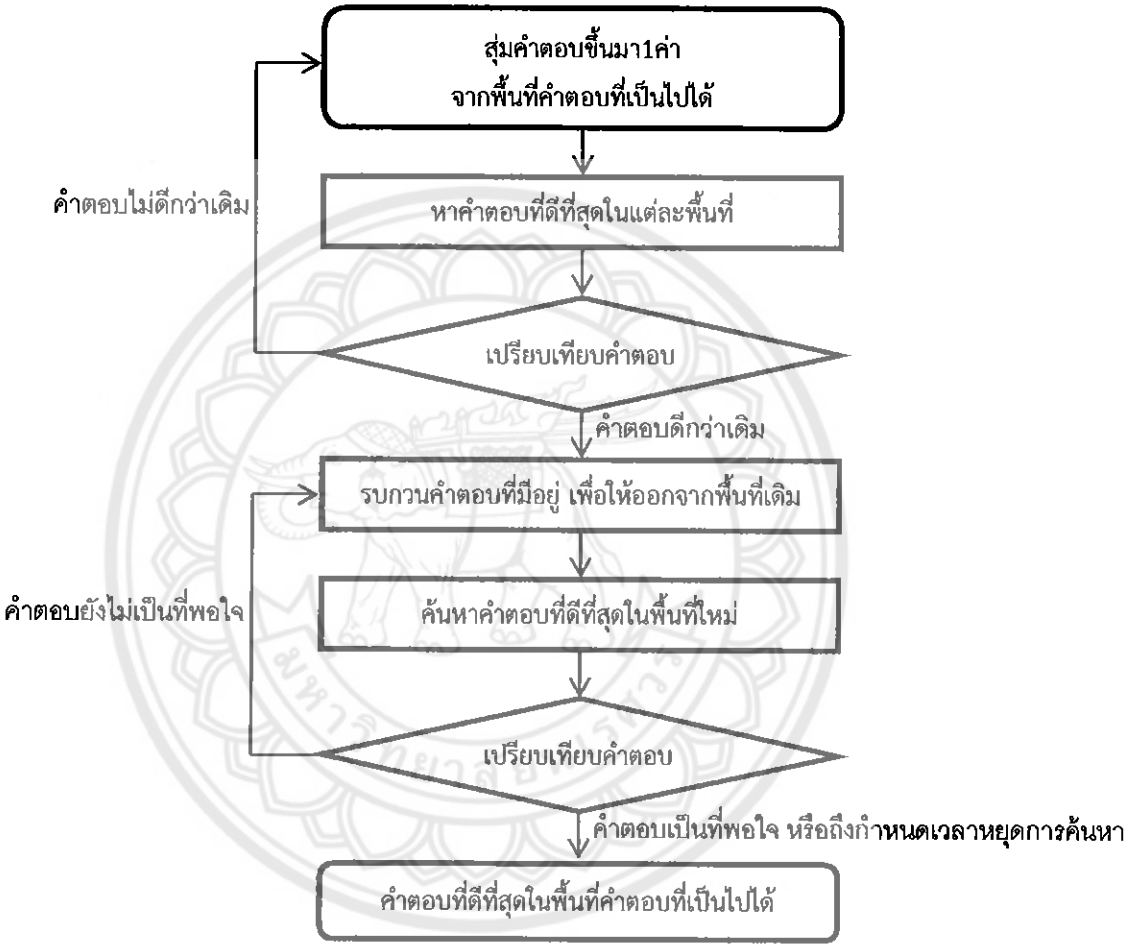
จากหลักการพื้นฐานดังกล่าว คุณภาพของเมตาฮิวริสติกที่ได้จากการประยุกต์ใช้การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำจะขึ้นอยู่กับสองประการหลัก

ประการแรก คือ คุณภาพของการค้นหาหรือปรับปรุงคำตอบเฉพาะที่ การใช้การค้นหาคำตอบเฉพาะที่ที่ดีจะทำให้หาค่าที่ดีที่สุดในแต่ละพื้นที่ได้รวดเร็วและได้ผลดี ถ้าใช้วิธีการค้นหาคำตอบเฉพาะที่ที่ไม่เหมาะสมจะมาสามารถหาคำตอบที่ดีที่สุดในแต่ละพื้นที่ได้ ทำให้ไม่ได้คำตอบที่เป็นคำตอบที่ดีที่สุดในพื้นที่ย่อย

ประการที่สอง คือ เทคนิคและการพัฒนาคำตอบที่มีอยู่ การพัฒนาคำตอบที่มีอย่างไม่มากไม่น้อยจนเกินไป ถ้ามากเกินไปจะทำให้ข้ามบางพื้นที่ที่มีคำตอบที่ดีไป เหมือนการหาอาหารของกบถ้ากระโดดไกลเกินไป อาจจะข้ามแหล่งที่มีแมลงชุกชุมที่ไม่ห่างจากพื้นที่ที่ค้นหาในปัจจุบัน ทำให้การค้นหาแบบนี้เรียกว่า การค้นหาแบบเดินสุ่ม (Random Work) ซึ่งจะเหมือนกับการเริ่มต้นหาคำตอบแรกเสมอ ไม่มีการเรียนรู้ของฮิวริสติก ซึ่งจะทำให้ได้คำตอบจากวิธีที่พัฒนาที่ไม่ดีตามที่คาดหวังไว้ แต่ถ้าการพัฒนาคำตอบที่มีอยู่น้อยเกินไปจะทำให้ไม่สามารถออกจากพื้นที่เก่าได้ เหมือนกับกบที่ขาหัก

กระโดดไปหาอาหารบริเวณอื่นไม่ได้ ต้องหาอาหารบริเวณเดิมๆ ไปเรื่อยๆ จนอาหารหมด ซึ่งก็จะได้คำตอบสุดท้ายที่เป็นเพียงคำตอบที่ดีที่สุดในพื้นที่ย่อยเท่านั้น (ระพีพันธ์, 2554)

การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ มีลำดับขั้นตอนการการทำงานเป็นไปตามที่ผู้ออกแบบได้กำหนดให้ทำการแก้ปัญหาอย่างใดอย่างหนึ่ง หรือหลายอย่าง ซึ่งขั้นตอนการทำงานนั้น ก็จะเปลี่ยนแปลงไปตามปัญหาตามความเหมาะสม ซึ่งจะมีขั้นตอนหลักๆ แสดงดังรูปที่ 2.11



รูปที่ 2.11 ลำดับขั้นตอนการทำงานโดยทั่วไปของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

นอกจากนี้การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำยังสามารถเขียนรหัสเทียมขึ้นมาใช้ได้ เพราะการเขียนขั้นตอนวิธีโดยการใช้ภาษาธรรมชาติอาจมีความกำกวมหรือยืดยาวเกินไป ทำให้เกิดการผิดพลาดได้ง่าย ซึ่งรหัสเทียมเป็นส่วนผสมของการใช้ภาษาธรรมชาติ และโปรแกรมภาษา เพื่อใช้ในการสื่อสารระหว่างผู้ใช้หลายกลุ่มที่มีความต้องการใช้ขั้นตอนวิธีต่างกันออกไป และรหัสเทียมยังไม่มีกฎในการเขียนตายตัว โดยมากขึ้นอยู่กับความถนัดของผู้ใช้ แต่มีข้อตกลงบางอย่างร่วมกันเป็นสากล ส่วนประกอบที่สำคัญของรหัสเทียม ได้แก่ ชื่อ คำสั่งกำหนดงาน คำสั่งควบคุม กลุ่มของคำสั่ง และข้อบันทึกหรือคำอธิบาย

หลักการเขียนรหัสเทียมซึ่งในที่นี้ จะกล่าวถึงหลักการเขียนรหัสเทียมโดยสังเขปได้ว่า 1. ถ้อยคำที่ใช้เขียนใช้ภาษาอังกฤษที่เข้าใจง่าย 2. ในหนึ่งบรรทัด ให้มีเพียงหนึ่งประโยคคำสั่ง 3. ใช้ย่อหน้าให้เป็นประโยชน์ ในการแสดงการควบคุมอย่างเป็นสัดส่วน 4. แต่ละประโยคคำสั่งให้เขียนจากบนลงล่าง และมีทางออกทางเดียว 5.กลุ่มของประโยคคำสั่งอาจรวมเป็นหมวดหมู่แล้วเรียกใช้เป็นโมดูลตัวอย่างของรหัสเทียมของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ แสดงดังรูปที่ 2.12

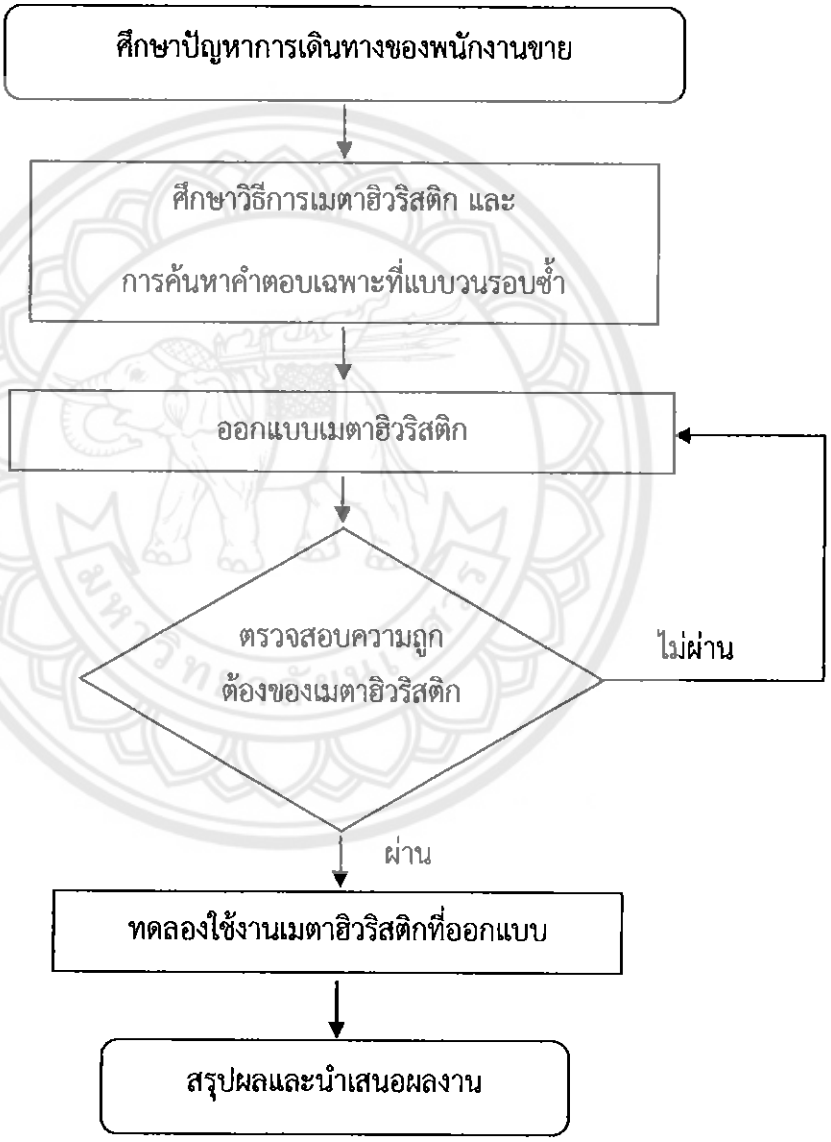
```
Input: ProblemInstance
Output: s*
s_0 ← ConstructInitialSolution();
s* ← LocalSearch(s_0);
while ¬ StopCondition() do
    s' ← Perturbation(s*, SearchHistory);
    s*' ← LocalSearch(s');
    s* ← AcceptanceCriterion(s*, s*', SearchHistory)
end
return s*;
```

รูปที่ 2.12 รหัสเทียมของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ
ที่มา : http://www.cleveralgorithms.com/nature-inspired/stochastic/iterated_local_search.html

จากรูปที่ 2.12 ค่า Input ถูกกำหนดให้เป็น Problem Instance หรือโจทย์ปัญหาที่จะทำการแก้ โดยมี Output เป็น s* หรือเป็นคำตอบที่ดีที่สุด (Best Solution) ที่ได้จากการแก้ปัญหาดังกล่าว ซึ่ง Output ที่ได้นี้จะประกอบไปด้วยขั้นตอนของ s_0 ที่เป็น Construct Initial Solution ( ); หรือการสร้างคำตอบเริ่มต้น (Initial Solution) และ s* ที่เป็น Local Search (s_0) หรือการค้นหาคำตอบเฉพาะที่ของคำตอบเริ่มต้น และจะกระทำไปต่อเรื่อยๆ จนกว่าจะถึงเงื่อนไขการหยุด หลังจากนั้นจะเป็นขั้นตอน s'ของขั้นตอนการทำ Perturbation (s*, Search History); หรือทำการรบกวนคำตอบที่มีอยู่ เพื่อให้ออกจากพื้นที่คำตอบเดิม และขั้นตอนของ s*' ซึ่งเป็นขั้นตอนของการทำ Local Search (s') หรือการค้นหาคำตอบที่ดีที่สุดในพื้นที่ใหม่ เมื่อได้คำตอบในพื้นที่ใหม่แล้ว จะทำการเปรียบเทียบคำตอบที่ใหม่ที่ได้กับคำตอบที่มีอยู่เดิม และถ้าหากค่าของคำตอบใหม่ที่ได้นั้น มีค่าถึงเกณฑ์ในการยอมรับของคำตอบที่ได้กำหนดไว้แล้วคำตอบที่ได้จะเป็นคำตอบที่ดีที่สุดในพื้นที่คำตอบที่เป็นไปได้ของพื้นที่ใหม่ (s*) แต่ถ้าคำตอบยังไม่เป็นที่น่าพอใจ สามารถนำคำตอบใหม่ที่ได้ กลับไปทำการรบกวนคำตอบอีกครั้ง และหาคำตอบที่ดีที่สุดในพื้นที่ใหม่ได้อีกเช่นกัน

บทที่ 3
วิธีการดำเนินโครงงาน

จากการศึกษาค้นคว้าข้อมูล และวิจัยเกี่ยวกับปัญหาการเดินทางของพนักงานขายดังที่กล่าวไว้ข้างต้นในบทที่ 1 และบทที่ 2 แล้ว และเพื่อทำการศึกษาวิจัยโครงงานที่เกี่ยวกับปัญหาการเดินทางของพนักงานขายนี้ จึงได้มีขั้นตอนการดำเนินโครงงานแสดงดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการดำเนินโครงงานของปัญหาการเดินทางของพนักงานขาย

3.1 ศึกษาปัญหาการเดินทางของพนักงานขาย

ในโครงงานนี้จะศึกษาปัญหาการเดินทางของพนักงานขายไปยังเมืองต่างๆ นั้นคือ ปัญหาที่มีการจัดเส้นทางการเดินทางจากจุดหนึ่งไปยังอีกจุดหนึ่งหรือหลายๆ จุด ซึ่งในการศึกษาปัญหานี้ก็เพื่อต้องการให้มีการเดินทางไปยังทุกๆ เมืองที่ได้มีการกำหนดไว้ จะต้องมีการเดินทางที่ดีที่สุด หรือมีระยะการเดินทางที่สั้นที่สุด โดยในการเดินทางของพนักงานขายจะใช้วิธีการเมตตาฮิวริสติก และการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

3.2 ศึกษาวิธีการเมตาฮิวริสติกและการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ

3.2.1 ศึกษาวิธีการเมตาฮิวริสติกที่มีชุดของลำดับขั้นการแก้ปัญหาแบบฮิวริสติกชนิดหนึ่ง โดยจะใช้การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำในการหาคำตอบที่ดีที่สุด

3.2.2 ศึกษาการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำที่พัฒนามาจากฮิวริสติกแบบค้นหาในพื้นที่บางส่วนในพื้นที่คำตอบ แล้วนำการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำมาประยุกต์ใช้สำหรับการแก้ปัญหาการเดินทางของพนักงานขาย ในโครงงานเล่มนี้ได้นำเสนอโจทย์ปัญหาการเดินทางของพนักงานขาย โดยการออกแบบเมตาฮิวริสติกที่สามารถช่วยในการหาเส้นทางการเดินทางมาประยุกต์ใช้กับวิธีการการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำซึ่งมุ่งหวังให้คำตอบที่ได้ออกมามีระยะการเดินทางนั้นๆ มีระยะทางที่สั้นที่สุดตามแบบที่กำหนด

3.3 ออกแบบเมตาฮิวริสติก

ในขั้นนี้จะกล่าวถึงการออกแบบเมตาฮิวริสติก โดยอาศัยขั้นตอนการทำงานของการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำมาประยุกต์ใช้ในการแก้ปัญหาการเดินทางของพนักงานขาย ซึ่งโครงงานนี้ได้ทำการออกแบบเมตาฮิวริสติกที่ผู้ใช้สามารถกำหนดจุดเริ่มต้น และจุดสิ้นสุดการเดินทาง และยังสามารถกำหนดเมืองที่จะเดินทางได้

3.4 ตรวจสอบความถูกต้องของการออกแบบเมตาฮิวริสติก

จากการที่ได้ออกแบบเมตาฮิวริสติกที่ประยุกต์เอาวิธีการการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำมาแก้ปัญหาการเดินทางของพนักงานขายเรียบร้อยแล้ว ต้องทำการตรวจสอบเมตาฮิวริสติก เพื่อความถูกต้องของขั้นตอนการทำงานของเมตาฮิวริสติกที่ป้อนข้อมูลเข้าไปเพื่อทำการทดสอบหรือสามารถคำนวณค่าของระยะทางการเดินทางของพนักงานที่สั้นที่สุดได้ หรือหาคำตอบที่ดีที่สุดได้

3.5 ทดลองแบบของเมตาฮิวริสติก

จากการตรวจสอบความถูกต้องของเมตาฮิวริสติกที่ออกแบบแล้ว ขั้นต่อไปจะนำเมตาฮิวริสติกที่ออกแบบมาเขียนลงบนโปรแกรมในคอมพิวเตอร์ที่ใช้ในการวิเคราะห์ข้อมูล โดยโปรแกรมที่ใช้ในการวิเคราะห์ข้อมูลต้องมีความสามารถในการคำนวณหาเส้นทางการเดินทางของพนักงานขายได้โดยอาศัยขั้นตอนการทำงานของวิธีการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ และทำการรันโปรแกรมที่มีเมตาฮิวริสติกที่ออกแบบ โดยการค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ ให้ได้คำตอบที่ออกมาตรงตามคำตอบที่เราต้องการ หรือให้ได้คำตอบที่ดีที่สุด หรือได้ระยะทางการเดินทางของพนักงานขายที่สั้นที่สุด

3.6 สรุปผลและนำเสนอผลงาน

การสรุปผลการทดลอง เป็นการสรุปผลลัพธ์ของการค้นหาเส้นทางที่ดีที่สุดที่แบบจำลองทางคณิตศาสตร์สามารถคำนวณหาได้ และทำการสรุปผลว่าแบบจำลองที่พัฒนาขึ้นมานั้นมีความสามารถในการใช้งานตรงตามเป้าหมายที่ตั้งไว้

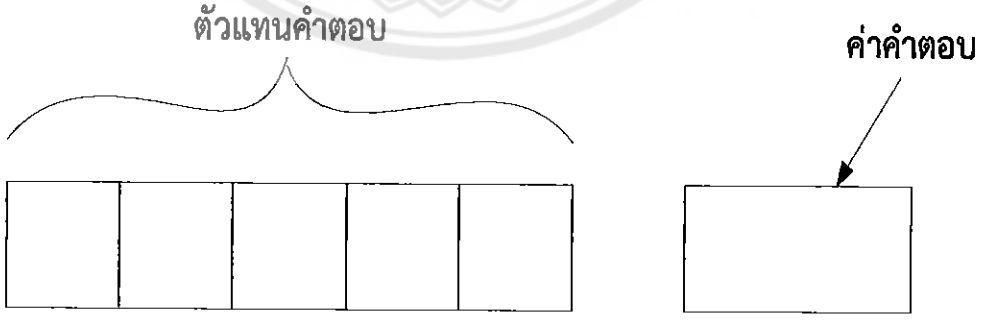
บทที่ 4
ผลการดำเนินโครงงาน

ในการดำเนินโครงงานการหาคำตอบของปัญหาการเดินทางของพนักงานขาย โดยใช้วิธีแบบเมตาฮิวริสติกนี้ ได้แบ่งเนื้อหาที่เกี่ยวข้องกับผลการทดลอง และการวิเคราะห์ออกเป็น 6 ส่วนหลัก โดยในส่วนแรกจะเป็นการสร้างตัวแทนคำตอบ และการหาค่าคำตอบในส่วนนี้จะทำให้ผู้อ่านเข้าใจถึงการใช้ตัวแทนคำตอบและการหาค่าคำตอบ ส่วนที่สองเป็นส่วนของออกแบบเมตาฮิวริสติก ในส่วนนี้จะข้อมูลที่ได้ศึกษามาทำการออกแบบ ส่วนที่สามวิธีการปรับปรุงคำตอบแบบสลับ (SWAP) และแบบเลื่อน (SLIDE) ในส่วนนี้จะทำให้ผู้ใช้เข้าใจการทำงานของวิธีการปรับปรุงคำตอบทั้งสอง ส่วนที่สี่เป็นส่วนของความสามารถของโปรแกรมและวิธีใช้อย่างย่อส่วนนี้จะอธิบายถึงความสามารถต่างๆ ของโปรแกรมที่โปรแกรมทำได้ ส่วนที่ห้าการออกแบบและผลการทดลองเพื่อเปรียบเทียบประสิทธิภาพของวิธีการหาคำตอบ ส่วนที่หกเป็นการวิเคราะห์ความแปรปรวน (ANOVA) ของผลลัพธ์โดยใช้โปรแกรม Minitab 16 โดยมีผลการทดลองและการวิเคราะห์ทั้ง 6 ส่วน ดังนี้

4.1 การสร้างตัวแทนคำตอบและการหาค่าคำตอบ

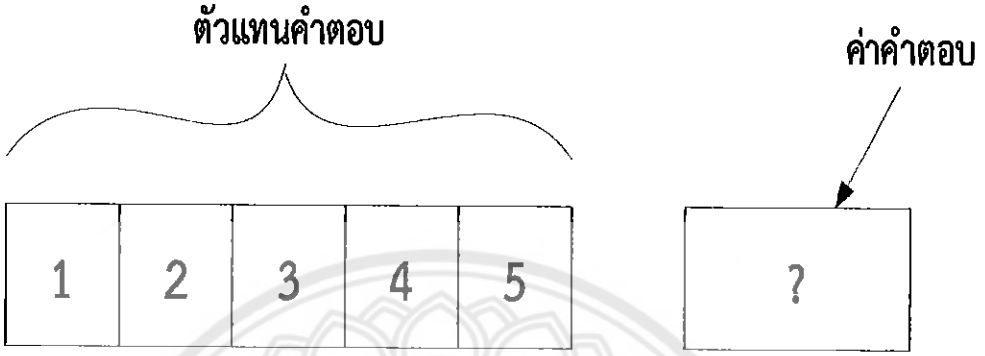
4.1.1 การสร้างตัวแทนคำตอบ

การใช้เมตาฮิวริสติก จำเป็นต้องมีการสร้างตัวแทนคำตอบ โดยตัวแทนคำตอบที่สร้างในโครงงานนี้ จะสร้างเป็นช่องๆ หรือเป็นตำแหน่ง ตามจำนวนของเมืองที่ผู้ใช้กำหนด หรือ n ตำแหน่ง (n คือ จำนวนเมือง) และจะมีตำแหน่งของค่าคำตอบที่แสดงค่าของระยะทางรวม n เมือง แสดงอยู่ทางด้านขวาของช่องตัวแทนคำตอบ n เมือง ดังรูปที่ 4.1 แสดงตัวแทนคำตอบ n = 5



รูปที่ 4.1 แสดงตัวแทนคำตอบแบบกำหนดให้ n = 5

ตัวอย่างที่ 1 กำหนดให้มีจำนวนเมือง 5 เมืองเรียงกัน และตัวแทนคำตอบนี้ ได้จากการสุ่มตัวเลขขึ้นมาหนึ่งชุด ในที่นี้กำหนดให้เป็น 1 – 2 – 3 – 4 – 5 ตามลำดับ ซึ่งหมายถึงการเดินทางของพนักงานขายจากเมือง 1 ไปยังเมืองที่ 2 เมือง 2 ไปยังเมือง 3 เมือง 3 ไปยังเมือง 4 เมือง 2 ไปยังเมือง 5 แล้วกลับมายังเมืองที่ 1 โดยจะมีการสร้างตัวแทนคำตอบดังรูปที่ 4.2



รูปที่ 4.2 ตัวอย่างการสร้างตัวแทนคำตอบ

4.1.2 การหาค่าคำตอบ

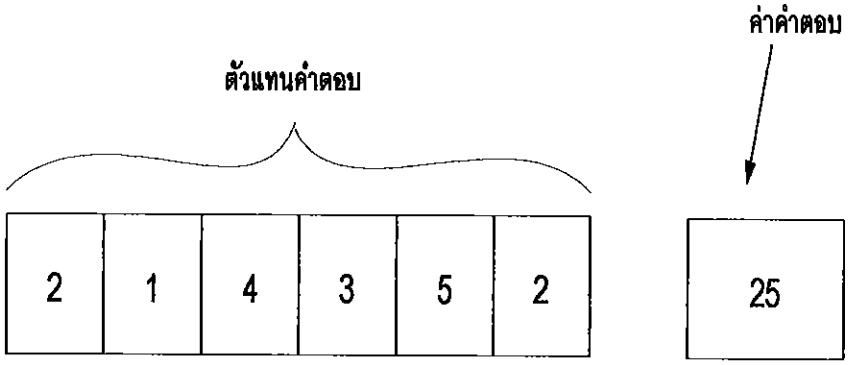
เมื่อทำการสร้างตัวแทนคำตอบเรียบร้อยแล้ว จากนั้นทำการหาค่าคำตอบของระยะทางรวมตั้งแต่เมืองที่ 1 ถึงเมืองที่ n

ตัวอย่างที่ 2 กำหนดให้มีจำนวนเมืองในการหาคำตอบ 5 เมือง และมีระยะทางระหว่างแต่ละเมือง ดังรูปที่ 4.3 โดยสมมติว่าตัวแทนคำตอบที่ต้องการหาค่าคำตอบเป็น 2 – 1 – 4 – 3 – 5 และกลับมายังเมืองเริ่มต้น

	1	2	3	4	5
1	0	6	7	5	3
2	4	0	9	4	10
3	5	8	0	2	8
4	9	1	6	0	1
5	7	2	10	3	0

รูปที่ 4.3 ตัวอย่างโจทย์การหาค่าคำตอบ

ในการคำนวณค่าคำตอบของระยะทางรวมจาก 2 – 1 – 4 – 3 – 5 - 2 ดังรูปที่ 4.4



รูปที่ 4.4 การคำนวณค่าคำตอบของระยะทางรวม

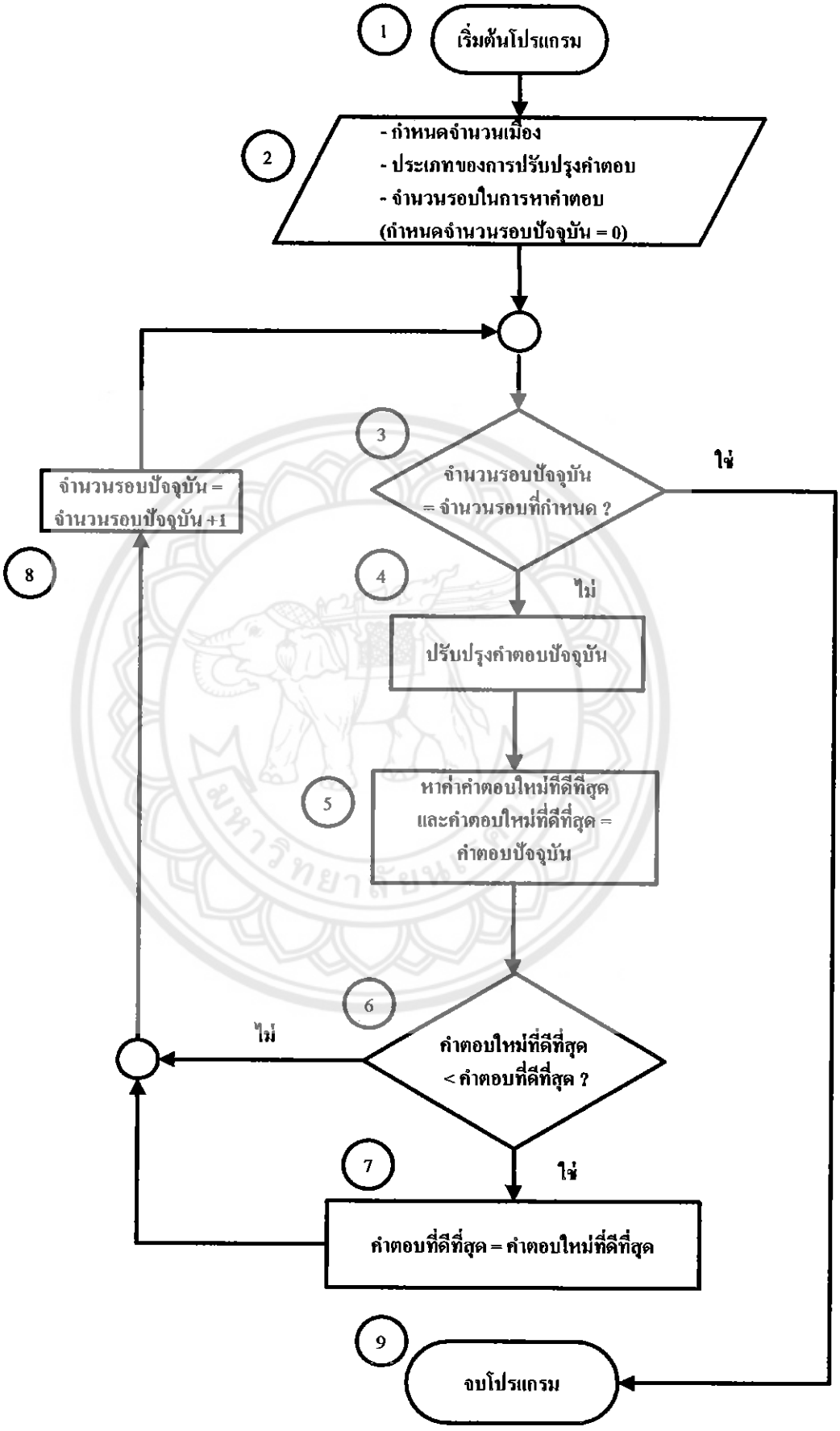
จารูปที่ 4.4 มีวิธีการคำนวณค่าคำตอบ ดังนี้ (ระยะทางจากเมืองที่ 2 ถึงเมืองที่ 1) + (ระยะทางจากเมืองที่ 1 ถึงเมืองที่ 4) + (ระยะทางจากเมืองที่ 4 ถึงเมืองที่ 3) + (ระยะทางจากเมืองที่ 3 ถึงเมืองที่ 5) + (ระยะทางจากเมืองที่ 5 ถึงเมืองที่ 2) = ค่าคำตอบของระยะทางรวม ค่าที่ได้คือ 4 + 5 + 6 + 8 + 2 = 25

4.2 ออกแบบเมตาฮิวริสติก

4.2.1 ออกแบบการทำงานของ Local Search Type 1

Local Search Type 1 คือ การหาคำตอบของปัญหาการเดินทางของพนักงานขาย โดยใช้การค้นหาคำตอบเฉพาะที่ (Local Search) ซึ่งมีวิธีการหาคำตอบ 2 วิธี คือ วิธีการปรับปรุงคำตอบแบบสลับ (SWAP) และวิธีการปรับปรุงคำตอบแบบเลื่อน (SLIDE) มีลักษณะในการหาคำตอบโดยหาค่าคำตอบใหม่ที่ดีที่สุดและนำค่าคำตอบใหม่ที่น้อยที่สุดมาแทนให้เป็นชุดคำตอบปัจจุบันทุกๆ รอบ โดยมีขั้นตอนการหาคำตอบ ดังรูปที่ 4.5

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร



รูปที่ 4.5 การทำงานของ Local Search Type 1

จากรูปที่ 4.5 จะเห็นว่ามีการทำงานของ Local Search Type 1 ตามขั้นตอน ดังนี้

4.2.1.1 ขั้นตอนที่ 1 เป็นการเริ่มต้นโปรแกรมจากหน้าต่างของโปรแกรมที่แสดงเมื่อเปิดโปรแกรมขึ้นมา

4.2.1.2 ขั้นตอนที่ 2 เป็นกำหนดจำนวนเมืองของปัญหาที่ต้องการที่จะให้โปรแกรมช่วยแก้ ว่ามีกี่เมือง จากนั้นเลือกวิธีการปรับปรุงคำตอบโดยมีให้เลือก 2 แบบ คือ SWAP หรือ SLIED จากนั้นกำหนดจำนวนรอบในการหาคำตอบ คือ จะให้โปรแกรมหาคำตอบกี่รอบ โปรแกรมจึงจะหยุดทำงาน

4.2.1.3 ขั้นตอนที่ 3 เป็นการเปรียบเทียบว่าจำนวนรอบปัจจุบันเท่ากับจำนวนรอบที่กำหนดหรือไม่ ถ้ายังไม่เท่าให้ไปทำขั้นตอนที่ 4 แต่ถ้าเท่ากันแล้วให้ทำขั้นตอนที่ 9

4.2.1.4 ขั้นตอนที่ 4 เป็นการปรับปรุงคำตอบเพื่อที่จะให้โปรแกรมหาคำตอบที่ดีที่สุดออกมา

4.2.1.5 ขั้นตอนที่ 5 เป็นการหาค่าคำตอบใหม่ที่ดีที่สุด และนำค่าคำตอบใหม่ที่ดีที่สุดมาเป็นค่าคำตอบปัจจุบัน

4.2.1.6 ขั้นตอนที่ 6 เป็นการเปรียบเทียบค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุดหรือไม่ ถ้าค่าคำตอบใหม่ที่ดีที่สุดไม่น้อยกว่าค่าคำตอบที่ดีที่สุด ให้ทำขั้นตอนที่ 8 แต่ถ้าค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุด ให้ทำขั้นตอนที่ 7

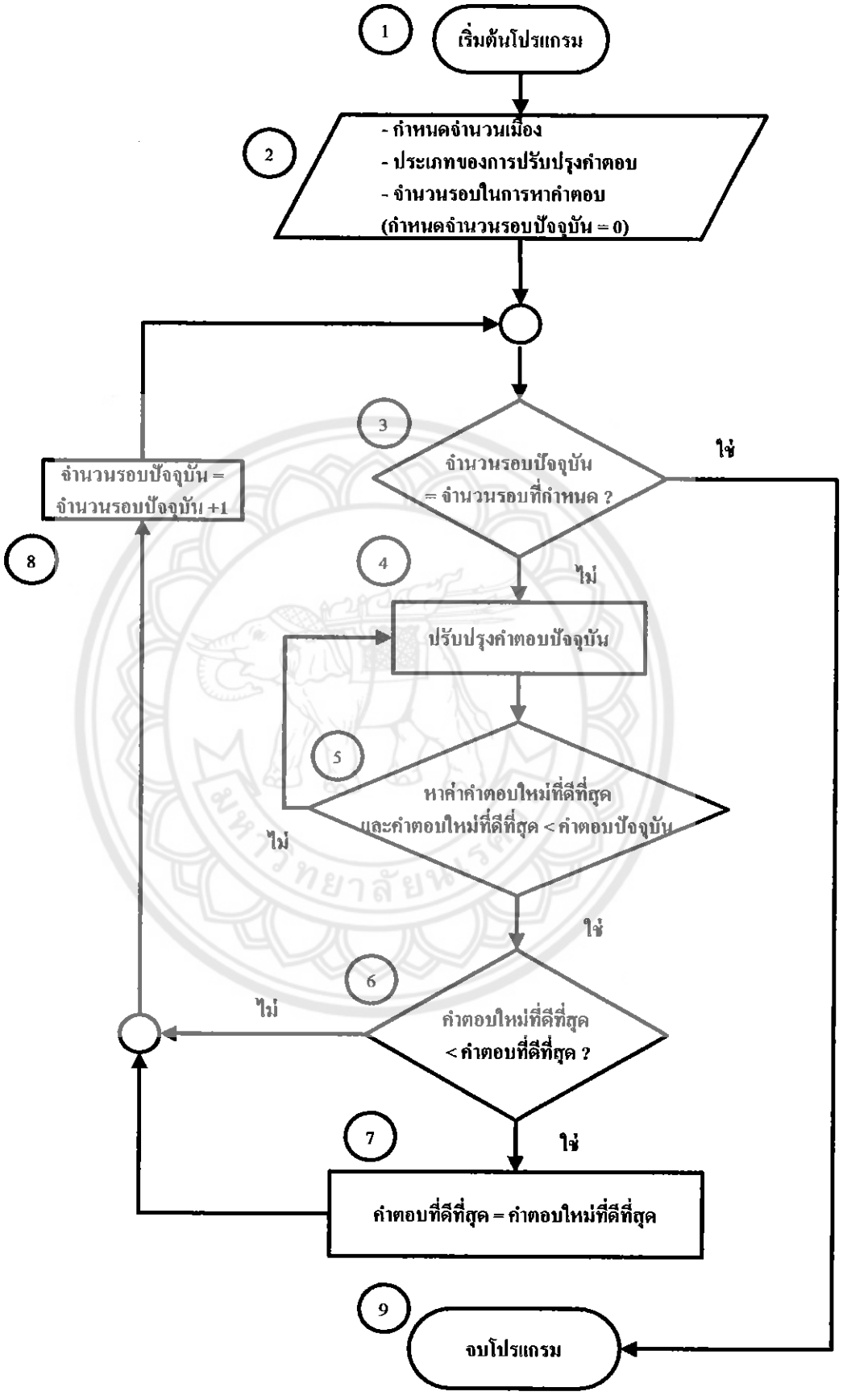
4.2.1.7 ขั้นตอนที่ 7 ถ้าค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุด ให้ค่าคำตอบใหม่ที่ดีที่สุดเป็นค่าคำตอบที่ดีที่สุดทันที

4.2.1.8 ขั้นตอนที่ 8 เป็นการให้จำนวนรอบปัจจุบันเท่ากับจำนวนรอบปัจจุบันบวก 1 แล้วจึงไปทำขั้นตอนที่ 3 (ให้จำนวนรอบปัจจุบันเริ่มต้นที่ 0)

4.2.1.9 ขั้นตอนที่ 9 เป็นการจบโปรแกรมในการจบโปรแกรมจะต้องมีจำนวนรอบปัจจุบันเท่ากับจำนวนรอบที่กำหนดถึงจะจบโปรแกรมได้

4.2.2 ออกแบบการทำงานของ Local Search Type 2

Local Search Type 2 คือ การหาคำตอบของปัญหาการเดินทางของพนักงานขาย โดยใช้การค้นหาคำตอบเฉพาะที่ (Local Search) ซึ่งมีวิธีการปรับปรุงคำตอบ 2 วิธี คือ วิธีการปรับปรุงคำตอบแบบสลับ และวิธีการปรับปรุงคำตอบแบบเลื่อน โดยวิธีการในการหาคำตอบด้วย Local Search Type 2 ถ้าค่าคำตอบใหม่ที่ดีที่สุดไม่น้อยกว่าค่าคำตอบปัจจุบันให้ทำการปรับปรุงคำตอบปัจจุบันต่อ มีลักษณะในการหาค่าคำตอบ โดยถ้าค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบปัจจุบันจะกำหนดให้ค่าคำตอบปัจจุบันเป็นค่าคำตอบที่ดีที่สุด โดยมีขั้นตอนการหาคำตอบ ดังรูปที่ 4.6



รูปที่ 4.6 การทำงานของ Local Search Type 2

จากรูปที่ 4.6 จะเห็นว่ามีการการทำงานของ Local Search Type 2 ตามขั้นตอน ดังนี้

4.2.1.1 ขั้นตอนที่ 1 เป็นการเริ่มต้นโปรแกรมจากหน้าต่างของโปรแกรมที่แสดงเมื่อเปิดโปรแกรมขึ้นมา

4.2.1.2 ขั้นตอนที่ 2 เป็นกำหนดจำนวนเมืองของปัญหาที่ต้องการที่จะให้โปรแกรมช่วยแก้ ว่ามีกี่เมือง จากนั้นเลือกวิธีการปรับปรุงคำตอบโดยมีให้เลือก 2 แบบ คือ SWAP หรือ SLIED จากนั้นกำหนดจำนวนรอบในการหาคำตอบ คือ จะให้โปรแกรมหาคำตอบกี่รอบ โปรแกรมจึงจะหยุดทำงาน

4.2.1.3 ขั้นตอนที่ 3 เป็นการเปรียบเทียบว่าจำนวนรอบปัจจุบันเท่ากับจำนวนรอบที่กำหนดหรือไม่ ถ้ายังไม่เท่าให้ไปทำขั้นตอนที่ 4 แต่ถ้าเท่ากันแล้วให้ทำขั้นตอนที่ 9

4.2.1.4 ขั้นตอนที่ 4 เป็นการปรับปรุงคำตอบเพื่อที่จะให้โปรแกรมหาคำตอบที่ดีที่สุดออกมา

4.2.2.5 ขั้นตอนที่ 5 เป็นการหาค่าคำตอบใหม่ที่ดีที่สุด และเปรียบเทียบค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบปัจจุบันหรือไม่ ถ้าค่าคำตอบใหม่ที่ดีที่สุดไม่น้อยกว่าค่าคำตอบปัจจุบันให้ขึ้นไปทำขั้นตอนที่ 4 ต่อ แต่ถ้าค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบปัจจุบันให้ไปทำขั้นตอนที่ 6

4.2.1.6 ขั้นตอนที่ 6 เป็นการเปรียบเทียบค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุดหรือไม่ ถ้าค่าคำตอบใหม่ที่ดีที่สุดไม่น้อยกว่าค่าคำตอบที่ดีที่สุด ให้ทำขั้นตอนที่ 8 แต่ถ้าค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุด ให้ทำขั้นตอนที่ 7

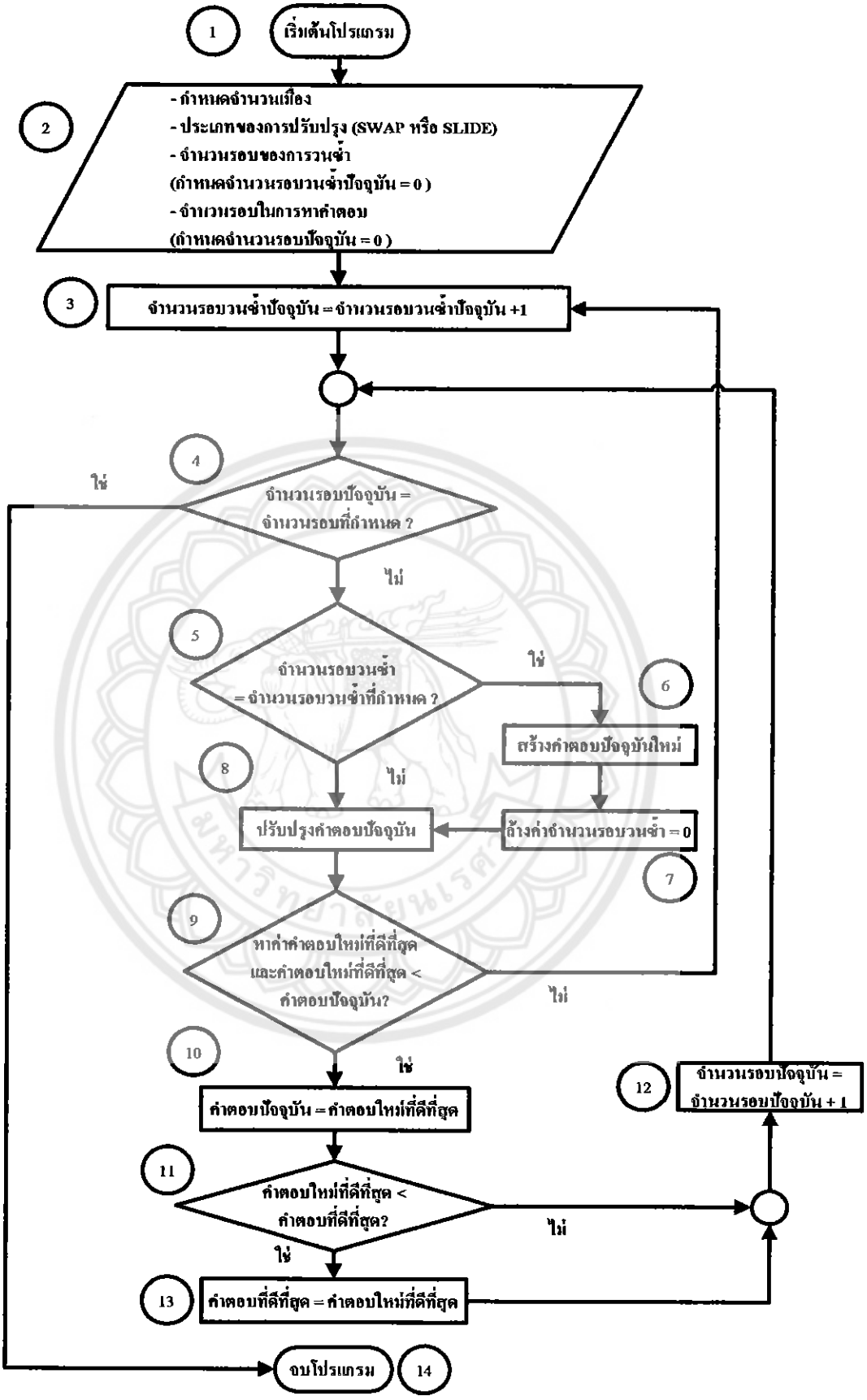
4.2.1.7 ขั้นตอนที่ 7 ถ้าค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุด ให้ค่าคำตอบใหม่ที่ดีที่สุดเป็นค่าคำตอบที่ดีที่สุดทันที

4.2.1.8 ขั้นตอนที่ 8 เป็นการให้จำนวนรอบปัจจุบันเท่ากับจำนวนรอบปัจจุบันบวก 1 แล้วจึงไปทำขั้นตอนที่ 3 (ให้จำนวนรอบปัจจุบันเริ่มต้นที่ 0)

4.2.1.9 ขั้นตอนที่ 9 เป็นการจบโปรแกรมในการจบโปรแกรมจะต้องมีจำนวนรอบปัจจุบันเท่ากับจำนวนรอบที่กำหนดถึงจะจบโปรแกรมได้

4.1.3 ออกแบบการทำงานของ Iterated Local Search

Iterated Local Search คือ การหาคำตอบของปัญหาการเดินทางของพนักงานขาย โดยใช้การค้นหาคำตอบเฉพาะที่แบบวนรอบซ้ำ (Iterated Local Search) ซึ่งมีวิธีการปรับปรุงคำตอบ 2 วิธี คือ วิธีการปรับปรุงคำตอบแบบสลับ และวิธีการปรับปรุงคำตอบแบบเลื่อน และยังมีการกำหนดค่าของการวนรอบซ้ำว่าไม่พบคำตอบที่ดีกว่าคำตอบเดิมกี่รอบจึงสุ่มหาชุดคำตอบใหม่โดยมีขั้นตอนการหาคำตอบ ดังรูปที่ 4.7



รูปที่ 4.7 การทำงานของ Iterated Local Search

จากรูปที่ 4.7 จะเห็นว่ามีการการทำงานของ Iterated Local Search ตามขั้นตอน ดังนี้

4.2.3.1 ขั้นตอนที่ 1 เป็นการเริ่มต้นโปรแกรมจากหน้าต่างของโปรแกรมที่แสดงเมื่อเปิดโปรแกรมขึ้นมา

4.2.3.2 ขั้นตอนที่ 2 เป็นการกำหนดจำนวนเมืองของปัญหาที่ต้องการที่จะให้โปรแกรมช่วยแก้ ว่ามีกี่เมือง จากนั้นเลือกวิธีการการปรับปรุงคำตอบโดยมีให้เลือก 2 แบบ คือ SWAP หรือ SLIED จากนั้นกำหนดจำนวนรอบในการหาคำตอบ คือ จะให้โปรแกรมหาคำตอบกี่รอบ โปรแกรมจึงจะหยุดทำงาน

4.2.3.3 ขั้นตอนที่ 3 เป็นการกำหนดให้จำนวนรอบวนซ้ำ เท่ากับจำนวนรอบวนซ้ำบวก 1 (ให้จำนวนรอบวนซ้ำปัจจุบันเริ่มต้นที่ 0)

4.2.3.12 ขั้นตอนที่ 12 เป็นการให้จำนวนรอบปัจจุบันเท่ากับจำนวนรอบปัจจุบันบวก 1 แล้วจึงไปทำขั้นตอนที่ 5 (ให้จำนวนรอบปัจจุบันเริ่มต้นที่ 0)

4.2.3.4 ขั้นตอนที่ 4 เป็นการเปรียบเทียบว่าจำนวนรอบปัจจุบันเท่ากับจำนวนรอบที่กำหนดหรือไม่ ถ้ายังไม่เท่าให้ขึ้นไปทำขั้นตอนที่ 5 แต่ถ้าเท่ากันให้ทำขั้นตอนที่ 14

4.2.3.5 ขั้นตอนที่ 5 เป็นการเปรียบเทียบว่าจำนวนรอบวนซ้ำเท่ากับจำนวนรอบวนซ้ำที่กำหนดหรือไม่ ถ้ายังไม่เท่าให้ขึ้นไปทำขั้นตอนที่ 6 แต่ถ้าเท่ากันให้ทำขั้นตอนที่ 8

4.2.3.6 ขั้นตอนที่ 6 เป็นการสร้างคำตอบปัจจุบันใหม่ที่เกิดจากการตรวจสอบขั้นตอนที่ 7

4.2.3.7 ขั้นตอนที่ 7 เป็นการล้างค่าจำนวนรอบวนซ้ำให้เป็น 0 แล้วให้ทำขั้นตอนที่ 8

4.2.3.8 ขั้นตอนที่ 8 เป็นการปรับปรุงคำตอบเพื่อที่จะให้โปรแกรมหาคำตอบที่ดีที่สุดออกมา

4.2.3.9 ขั้นตอนที่ 9 เป็นการหาค่าคำตอบใหม่ที่ดีที่สุดและตรวจสอบค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบปัจจุบันหรือไม่ ถ้าค่าคำตอบใหม่ที่ดีที่สุดไม่น้อยกว่าค่าคำตอบปัจจุบันให้ไปขั้นตอนที่ 3 แต่ถ้าค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบปัจจุบันให้ทำขั้นตอนที่ 10

4.2.3.10 ขั้นตอนที่ 10 ค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบปัจจุบัน ให้ค่าคำตอบใหม่ที่ดีที่สุดเป็นค่าคำตอบปัจจุบันทันที

4.2.3.11 ขั้นตอนที่ 11 เป็นการเปรียบเทียบค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุดหรือไม่ ถ้าค่าคำตอบใหม่ที่ดีที่สุดไม่น้อยกว่าค่าคำตอบที่ดีที่สุด ให้ทำขั้นตอนที่ 12 แต่ถ้าค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุด ให้ทำขั้นตอนที่ 13

4.2.3.13 ขั้นตอนที่ 13 ค่าคำตอบใหม่ที่ดีที่สุดน้อยกว่าค่าคำตอบที่ดีที่สุด ให้ค่าคำตอบใหม่ที่ดีที่สุดเป็นค่าคำตอบที่ดีที่สุดทันที

4.2.3.14 ขั้นตอนที่ 14 เป็นการจบโปรแกรม ในการจบโปรแกรมจะต้องมีจำนวนรอบปัจจุบันเท่ากับจำนวนรอบที่กำหนดถึงจะจบโปรแกรมได้

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

4.2.4.1 Local Search Type 1 มีลักษณะในการหาคำตอบโดยหาค่าคำตอบใหม่ที่ดีที่สุดและนำค่าคำตอบใหม่ที่น้อยที่สุดมาแทนให้เป็นชุดคำตอบปัจจุบันทุกๆ รอบ

4.2.4.2 Local Search Type 2 มีลักษณะในการหาคำตอบโดยหาค่าคำตอบใหม่ที่ดีที่สุดและเปรียบเทียบว่าค่าคำตอบใหม่ที่น้อยที่สุดน้อยกว่าค่าคำตอบปัจจุบันหรือไม่ วิธีนี้มีความแตกต่างจาก Local Search Type 1 คือ ถ้าค่าคำตอบใหม่ที่น้อยที่สุดไม่น้อยกว่าค่าคำตอบปัจจุบันให้ทำการปรับปรุงคำตอบปัจจุบันต่อ แต่ถ้าค่าคำตอบใหม่ที่น้อยที่สุดน้อยกว่าค่าคำตอบปัจจุบัน จะกำหนดให้ค่าคำตอบปัจจุบันเป็นค่าคำตอบที่ดีที่สุด

4.2.4.3 Iterated Local Search มีลักษณะในการหาคำตอบโดยหาค่าคำตอบใหม่ที่ดีที่สุดและเปรียบเทียบว่าค่าคำตอบใหม่ที่น้อยที่สุดน้อยกว่าค่าคำตอบปัจจุบันหรือไม่ วิธีนี้มีความแตกต่างจาก Local Search Type 1 คือ ถ้าค่าคำตอบใหม่ที่น้อยที่สุดไม่น้อยกว่าค่าคำตอบปัจจุบันให้ทำการปรับปรุงคำตอบปัจจุบันต่อ แต่ถ้าค่าคำตอบใหม่ที่น้อยที่สุดน้อยกว่าค่าคำตอบปัจจุบัน จะกำหนดให้ค่าคำตอบปัจจุบันเป็นค่าคำตอบที่ดีที่สุด และมีความแตกต่างจาก Local Search Type 2 คือ ถ้าค่าคำตอบใหม่ที่น้อยที่สุดไม่น้อยกว่าค่าคำตอบปัจจุบันตามจำนวนรอบการวนซ้ำที่กำหนด Iterated Local Search จะสุ่มหาค่าคำตอบปัจจุบันใหม่

4.3 วิธีการปรับปรุงคำตอบแบบสลับ (SWAP) และแบบเลื่อน (SLIDE)

4.3.1 วิธีการปรับปรุงคำตอบแบบสลับ (SWAP)

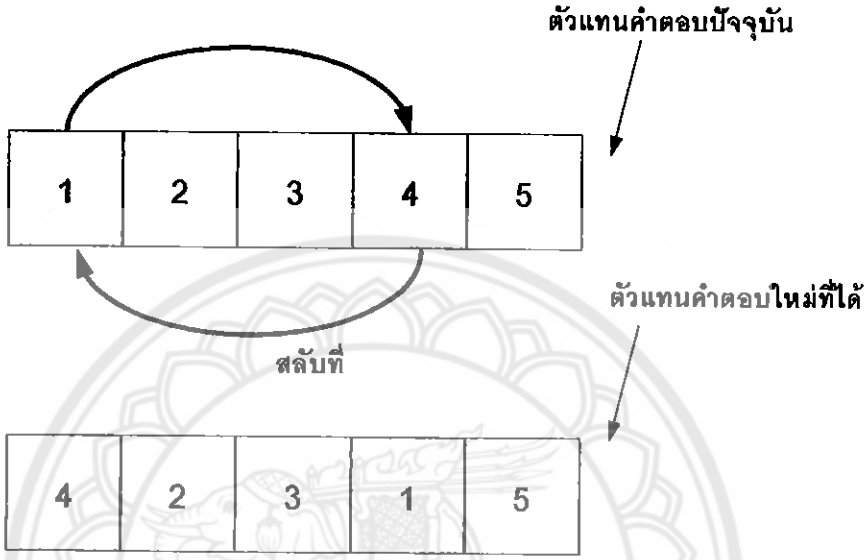
วิธีการปรับปรุงคำตอบแบบสลับ คือ วิธีการปรับปรุงคำตอบโดยการนำเอาตัวแทนคำตอบปัจจุบันมาสลับตำแหน่งกัน ซึ่งการสลับตัวแทนคำตอบในแต่ละครั้งจะมีการสุ่มช่องตัวแทนคำตอบเพื่อที่จะสลับตัวแทนคำตอบในการสลับที่ของตัวแทนคำตอบจะเริ่มต้นให้ช่องที่ 1 ทางด้านซ้ายของตัวแทนคำตอบเป็นตัวหลักแล้วจึงสุ่มหาช่องตัวแทนคำตอบที่จะสลับกัน และในรอบของการสลับถัดๆ มาจะให้ช่องตัวแทนคำตอบที่ 2, 3, ..., n (n คือ จำนวนเมืองที่กำหนดในตัวแทนคำตอบ) เป็นตัวสุ่มหาช่องตัวแทนคำตอบที่จะสลับกัน โดยกระบวนการของการสลับ สามารถอธิบายได้ตามตัวอย่างดังต่อไปนี้

4.3.1.1 สร้างตัวแทนคำตอบปัจจุบันขึ้นมา สมมติให้ตัวแทนคำตอบปัจจุบัน ที่จะพิจารณาหรือจะปรับปรุงสุ่มได้ 1 – 2 – 3 – 4 - 5 เรียงกัน ดังรูปที่ 4.8

1	2	3	4	5

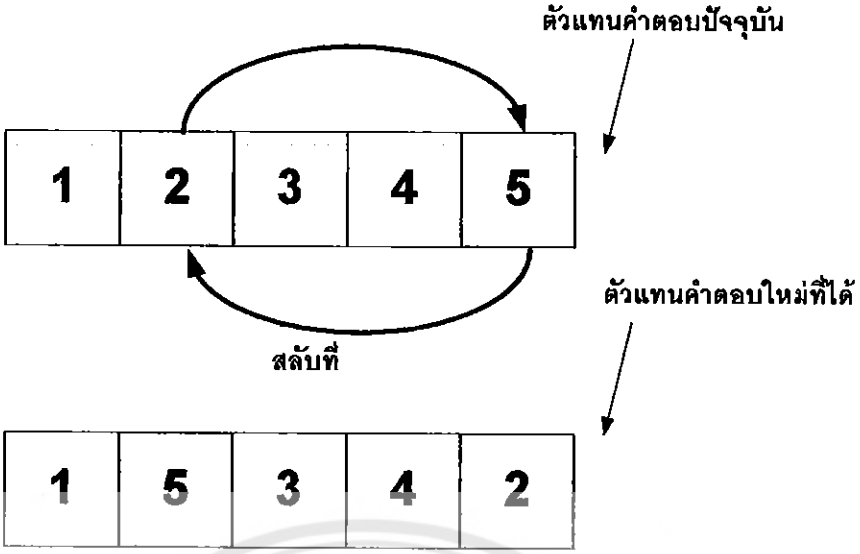
รูปที่ 4.8 ตัวแทนคำตอบปัจจุบันแบบสลับ

4.3.1.2 จากนั้นกำหนดให้ตัวแทนคำตอบตำแหน่งที่ 1 เป็นตัวหลักแล้วทำการสุ่มเพื่อที่จะสลับกับตัวแทนคำตอบตำแหน่งอื่น สมมติให้ตัวแทนคำตอบตำแหน่งที่ 1 สุ่มไปเจอกับตัวแทนคำตอบช่องที่ 4 และทำการสลับที่กันระหว่างตัวแทนคำตอบตำแหน่งที่ 1 สลับกับตัวแทนคำตอบตำแหน่งที่ 4 ตัวแทนคำตอบที่สลับกันเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.9



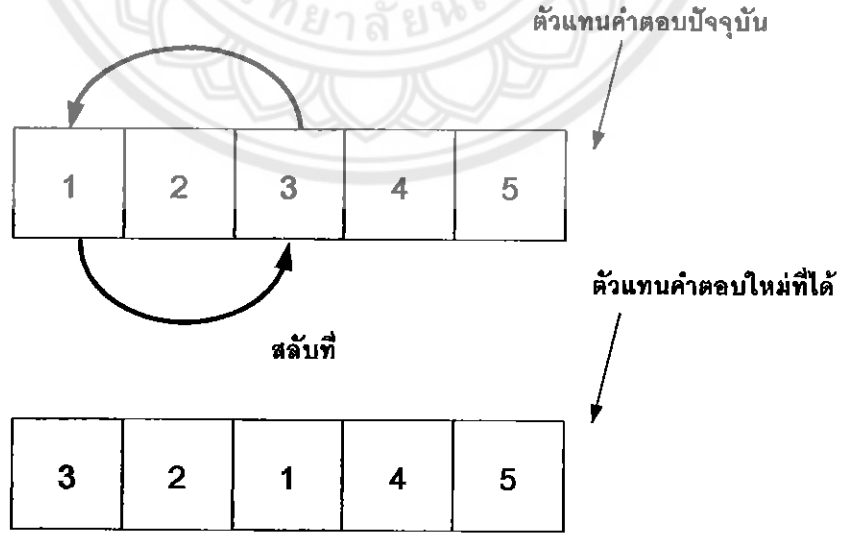
รูปที่ 4.9 การสลับโดยยึดตำแหน่ง 1 เป็นตัวหลัก

4.3.1.3 เมื่อทำการสลับตัวแทนคำตอบปัจจุบันในรอบแรกแล้ว จากนั้นในรอบต่อมาจะนำตัวแทนคำตอบปัจจุบันที่ได้จากรูปที่ 4.8 มาทำการสลับใหม่โดยกำหนดให้ตัวแทนคำตอบตำแหน่งที่ 2 เป็นตัวหลักแล้วทำการสุ่มเพื่อที่จะสลับกับตัวแทนคำตอบตำแหน่งอื่น สมมติให้ตัวแทนคำตอบตำแหน่งที่ 2 สุ่มไปเจอกับตัวแทนคำตอบตำแหน่งที่ 5 และทำการสลับที่กันระหว่างตัวแทนคำตอบตำแหน่งที่ 2 สลับกับตัวแทนคำตอบตำแหน่งที่ 5 ตัวแทนคำตอบที่สลับกันเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.10



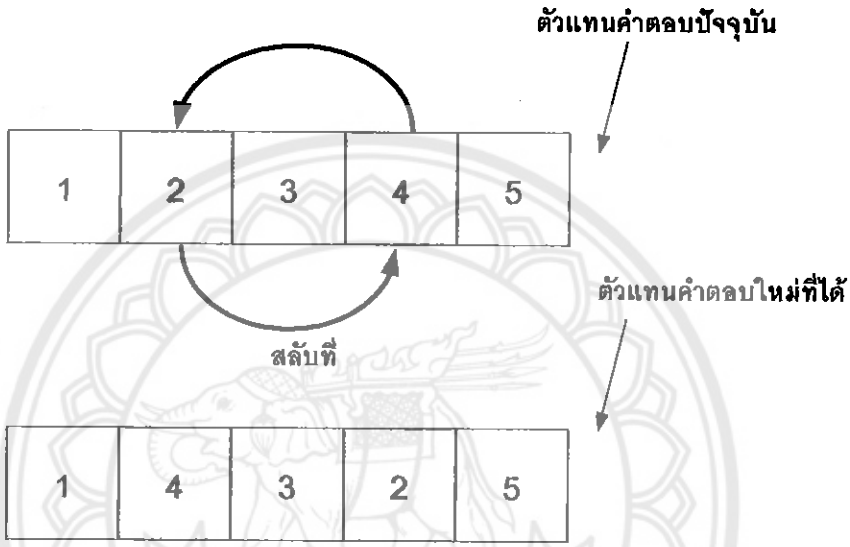
รูปที่ 4.10 การสลับโดยยึดตำแหน่ง 2 เป็นตัวหลัก

4.3.1.4 เมื่อทำการสลับตัวแทนคำตอบปัจจุบันในรอบที่สองแล้ว จากนั้นในรอบต่อมาจะนำตัวแทนคำตอบปัจจุบันที่ได้จากรูปที่ 4.8 มาทำการสลับใหม่โดยกำหนดให้ตัวแทนคำตอบตำแหน่งที่ 3 เป็นตัวหลักแล้วทำการสุ่มเพื่อที่จะสลับกับตัวแทนคำตอบตำแหน่งอื่น สมมติให้ตัวแทนคำตอบตำแหน่งที่ 3 สุ่มไปเจอกับตัวแทนคำตอบตำแหน่งที่ 1 และทำการสลับที่กันระหว่างตัวแทนคำตอบตำแหน่งที่ 3 สลับกับตัวแทนคำตอบตำแหน่งที่ 1 ตัวแทนคำตอบที่สลับกันเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.11



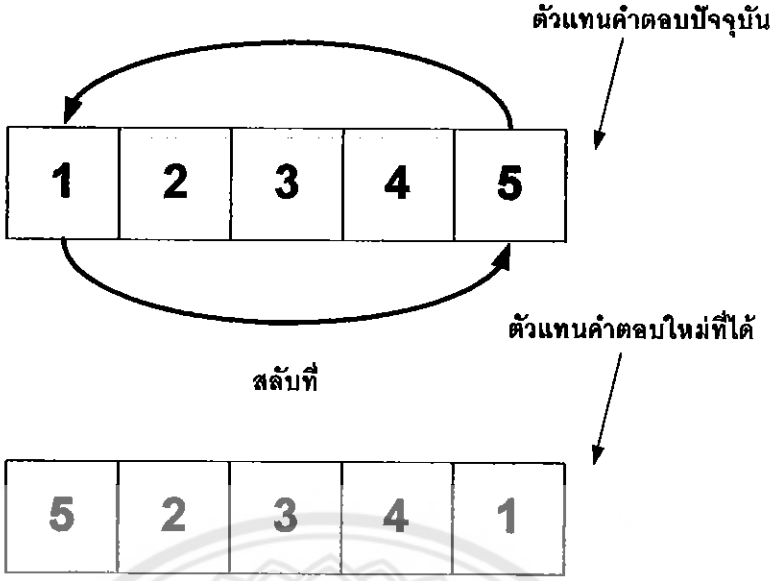
รูปที่ 4.11 การสลับโดยยึดตำแหน่ง 3 เป็นตัวหลัก

4.3.1.5 เมื่อทำการสลับตัวแทนคำตอบปัจจุบันในรอบที่สามแล้ว จากนั้นในรอบต่อมาจะนำตัวแทนคำตอบปัจจุบันที่ได้จากรูปที่ 4.8 มาทำการสลับใหม่โดยกำหนดให้ตัวแทนคำตอบตำแหน่งที่ 4 เป็นตัวหลักแล้วทำการสุ่มเพื่อที่จะสลับกับตัวแทนคำตอบตำแหน่งอื่น สมมติให้ตัวแทนคำตอบตำแหน่งที่ 4 สุ่มไปเจอกับตัวแทนคำตอบตำแหน่งที่ 2 และทำการสลับที่กันระหว่างตัวแทนคำตอบตำแหน่งที่ 4 สลับกับตัวแทนคำตอบตำแหน่งที่ 2 ตัวแทนคำตอบที่สลับกันเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.12



รูปที่ 4.12 การสลับโดยยึดตำแหน่ง 4 เป็นตัวหลัก

4.3.1.6 เมื่อทำการสลับตัวแทนคำตอบปัจจุบันในรอบที่สี่แล้ว จากนั้นในรอบต่อมาจะนำตัวแทนคำตอบปัจจุบันที่ได้จากรูปที่ 4.8 มาทำการสลับใหม่โดยกำหนดให้ตัวแทนคำตอบตำแหน่งที่ 5 เป็นตัวหลักแล้วทำการสุ่มเพื่อที่จะสลับกับตัวแทนคำตอบตำแหน่งอื่น สมมติให้ตัวแทนคำตอบตำแหน่งที่ 5 สุ่มไปเจอกับตัวแทนคำตอบตำแหน่งที่ 1 และทำการสลับที่กันระหว่างตัวแทนคำตอบตำแหน่งที่ 5 สลับกับตัวแทนคำตอบตำแหน่งที่ 1 ตัวแทนคำตอบที่สลับกันเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.13



รูปที่ 4.13 การสลับโดยยึดตำแหน่ง 5 เป็นตัวหลัก

4.3.1.7 จากการที่ได้ทำการสลับดังตัวอย่างข้างต้นจะเห็นได้ว่าจะมี**ตัวแทนคำตอบใหม่ที่ได้เกิดขึ้น** ได้แก่

ก. 4 – 2 – 3 – 1 - 5

ข. 1 – 5 – 3 – 4 - 2

ค. 3 – 2 – 1 – 4 - 5

ง. 1 – 4 – 3 – 2 - 5

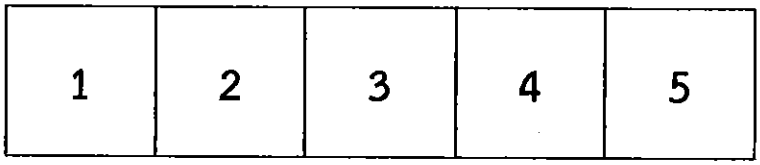
จ. 5 – 2 – 3 – 4 - 1

สรุป ในการสลับที่แต่ละครั้งหรือแต่ละรอบจะมีตัวแทนคำตอบใหม่ที่ได้ครั้งละ 1 คำตอบ หรือถ้ามีเมืองจำนวน n เมือง ตัวแทนคำตอบใหม่ที่ได้จะมี n ตัวแทนคำตอบใหม่

4.3.2 วิธีการปรับปรุงคำตอบแบบเลื่อน (SLIDE)

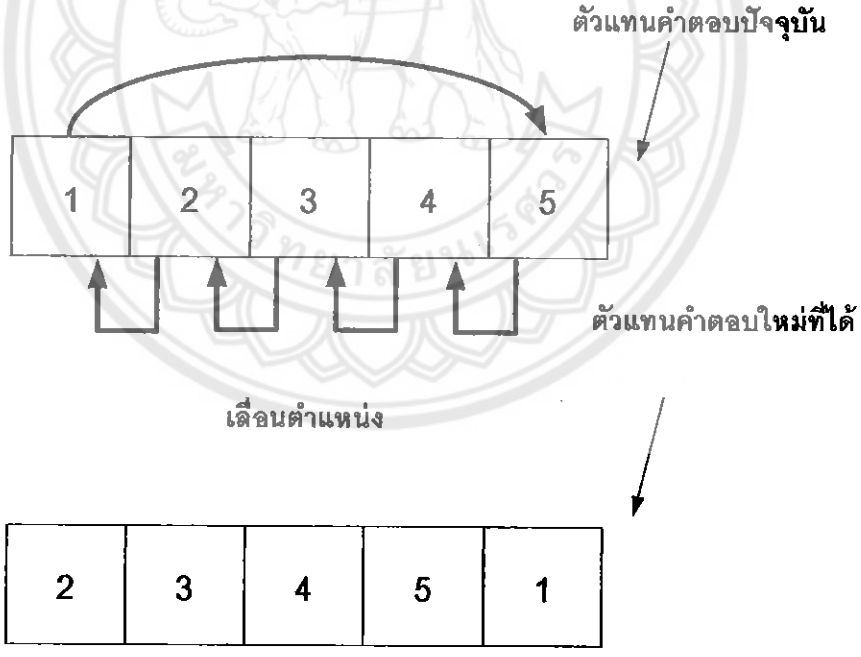
วิธีการปรับปรุงคำตอบแบบเลื่อน คือ วิธีการปรับปรุงคำตอบโดยการนำเอาตัวแทนคำตอบปัจจุบันมาเลื่อนเพื่อหาคำตอบ ซึ่งการเลื่อนตัวแทนคำตอบในแต่ละครั้งจะมีการสุ่มช่องตัวแทนคำตอบเพื่อที่จะเลื่อนตัวแทนคำตอบ โดยในการเลื่อนช่องของตัวแทนคำตอบจะเริ่มต้นให้ตำแหน่งที่ 1 ทางด้านซ้ายของตัวแทนคำตอบเป็นตัวหลักแล้วจึงสุ่มช่องตัวแทนคำตอบเพื่อที่จะเลื่อนตัวแทนคำตอบ และในรอบของการเลื่อนถัดๆ มาจะให้ช่องตัวแทนคำตอบที่ 2, 3, ..., n (n คือจำนวนเมืองที่กำหนดในตัวแทนคำตอบ) เป็นตัวสุ่มช่องตัวแทนคำตอบเพื่อที่จะเลื่อนตัวแทนคำตอบ โดยกระบวนการของการเลื่อน สามารถอธิบายได้ตามตัวอย่างดังต่อไปนี้

4.3.2.1 สร้างตัวแทนคำตอบปัจจุบันขึ้นมา สมมติให้ตัวแทนคำตอบปัจจุบัน ที่จะพิจารณาหรือจะปรับปรุงสุ่มได้ 1 – 2 – 3 – 4 - 5 เรียงกัน ดังรูปที่ 4.14



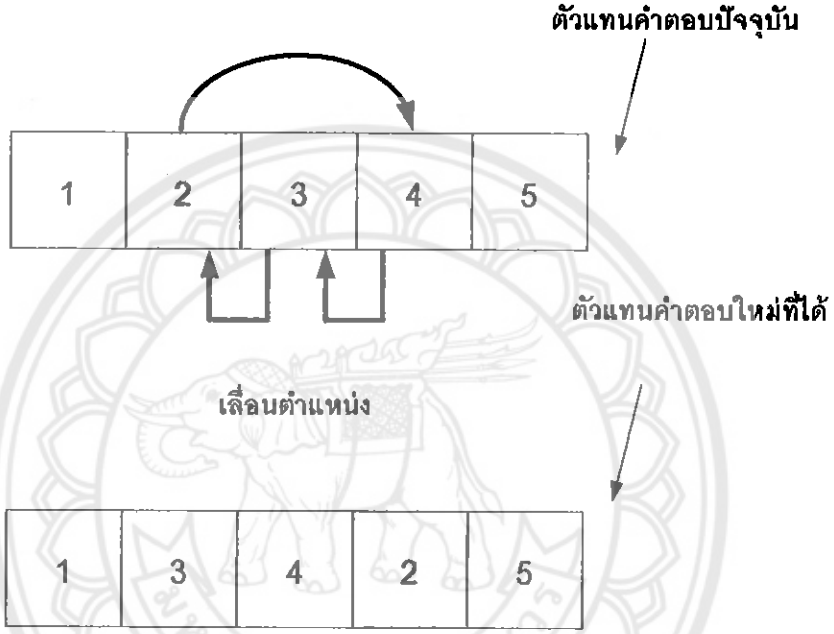
รูปที่ 4.14 ตัวแทนคำตอบปัจจุบันแบบเลื่อน

4.3.2.2 จากนั้นกำหนดให้ตัวแทนคำตอบตำแหน่งที่ 1 เป็นตัวหลักที่จะต้องทำการสุ่มหาช่องเพื่อที่จะเลื่อน สมมติให้ตัวแทนคำตอบตำแหน่งที่ 1 สุ่มไปเจอกับตัวแทนคำตอบตำแหน่งที่ 5 และนำเอาตัวแทนคำตอบตำแหน่งที่ 1 ไปแทนในช่องคำแทนคำตอบตำแหน่งที่ 5 จากนั้นนำตัวแทนคำตอบตำแหน่งที่ 2 แทนใน 1 ตัวแทนคำตอบตำแหน่งที่ 3 แทนใน 2 ตัวแทนคำตอบตำแหน่งที่ 4 แทนใน 3 ตัวแทนคำตอบตำแหน่งที่ 5 แทนใน 4 เมื่อเลื่อนเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.15



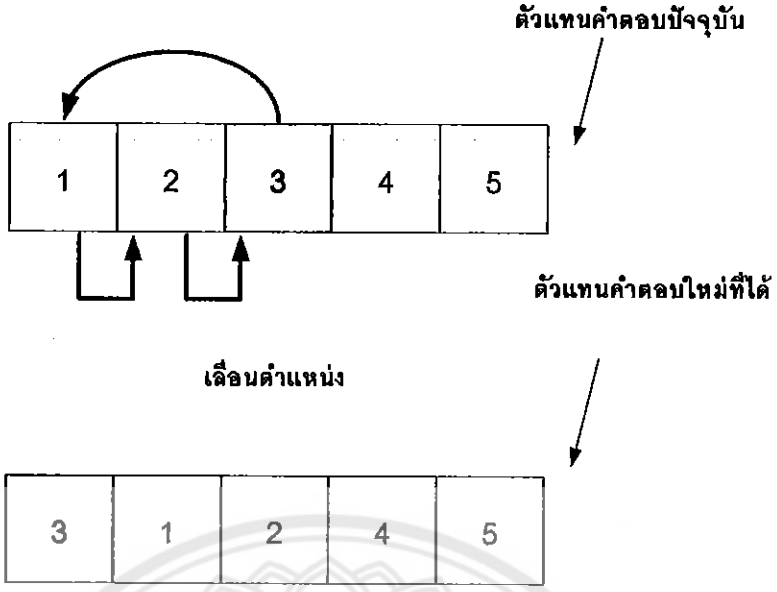
รูปที่ 4.15 การเลื่อนโดยยึดตำแหน่ง 1 เป็นตัวหลัก

4.3.2.3 เมื่อทำการเลื่อนตัวแทนคำตอบเริ่มต้นในรอบแรกแล้ว จากนั้นในรอบต่อมาจะนำตัวแทนคำตอบปัจจุบันที่ได้จากรูปที่ 4.14 มาทำการเลื่อนใหม่ โดยให้ตัวแทนคำตอบตำแหน่งที่ 2 เป็นตัวหลักที่จะต้องทำการสุ่มหาตำแหน่งเพื่อที่จะเลื่อน สมมติให้ตัวแทนคำตอบตำแหน่งที่ 2 สุ่มไปเจอกับตัวแทนคำตอบตำแหน่งที่ 4 และนำเอาตัวแทนคำตอบตำแหน่งที่ 2 ไปแทนในช่องคำแทนคำตอบตำแหน่งที่ 4 จากนั้นนำตัวแทนคำตอบตำแหน่งที่ 3 แทนใน 2 ตัวแทนคำตอบตำแหน่งที่ 4 แทนใน 3 เมื่อเลื่อนเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.16



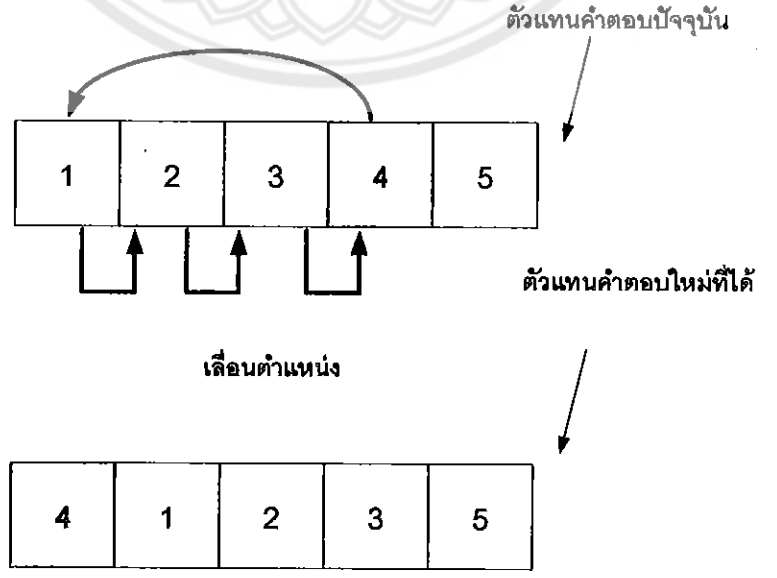
รูปที่ 4.16 การเลื่อนโดยยึดตำแหน่ง 2 เป็นตัวหลัก

4.3.2.4 เมื่อทำการเลื่อนตัวแทนคำตอบเริ่มต้นในรอบที่สองแล้ว จากนั้นในรอบต่อมาจะนำตัวแทนคำตอบปัจจุบันที่ได้จากรูปที่ 4.14 มาทำการเลื่อนใหม่ โดยให้ตัวแทนคำตอบตำแหน่งที่ 3 เป็นตัวหลักที่จะต้องทำการสุ่มหาตำแหน่งเพื่อที่จะเลื่อน สมมติให้ตัวแทนคำตอบตำแหน่งที่ 3 สุ่มไปเจอกับตัวแทนคำตอบตำแหน่งที่ 1 และนำเอาตัวแทนคำตอบตำแหน่งที่ 3 ไปแทนในช่องคำแทนคำตอบตำแหน่งที่ 1 จากนั้นนำตัวแทนคำตอบตำแหน่งที่ 1 แทนใน 2 ตัวแทนคำตอบตำแหน่งที่ 2 แทนใน 3 เมื่อเลื่อนเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.17



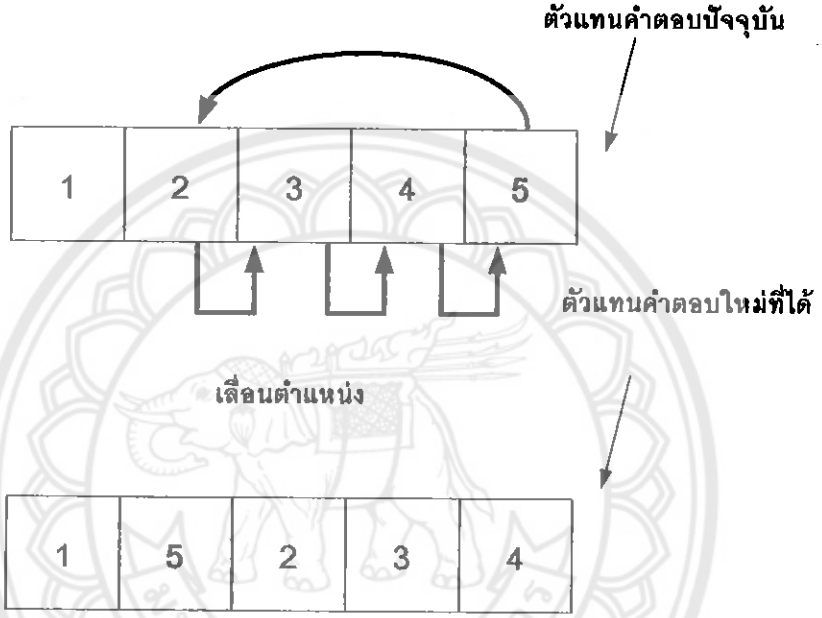
รูปที่ 4.17 การเลื่อนโดยยึดตำแหน่ง 3 เป็นตัวหลัก

4.3.2.5 เมื่อทำการเลื่อนตัวแทนคำตอบเริ่มต้นในรอบที่สามแล้ว จากนั้นในรอบต่อมาจะนำตัวแทนคำตอบปัจจุบันที่ได้จากรูปที่ 4.14 มาทำการเลื่อนใหม่ โดยให้ตัวแทนคำตอบตำแหน่งที่ 4 เป็นตัวหลักที่จะต้องทำการสุ่มหาช่องเพื่อที่จะเลื่อน สมมติให้ตัวแทนคำตอบตำแหน่งที่ 4 สุ่มไปเจอกับตัวแทนคำตอบตำแหน่งที่ 1 และนำเอาตัวแทนคำตอบตำแหน่งที่ 4 ไปแทนในช่องคำแทนคำตอบตำแหน่งที่ 1 จากนั้นนำตัวแทนคำตอบตำแหน่งที่ 1 แทนใน 2 ตัวแทนคำตอบตำแหน่งที่ 2 แทนใน 3 ตัวแทนคำตอบตำแหน่งที่ 3 แทนใน 4 เมื่อเลื่อนเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.18



รูปที่ 4.18 การเลื่อนโดยยึดตำแหน่ง 4 เป็นตัวหลัก

4.3.2.6 เมื่อทำการเลื่อนตัวแทนคำตอบเริ่มต้นในรอบที่สี่แล้ว จากนั้นในรอบต่อมาจะนำตัวแทนคำตอบปัจจุบันที่ได้จากรูปที่ 4.14 มาทำการเลื่อนใหม่ โดยให้ตัวแทนคำตอบตำแหน่งที่ 5 เป็นตัวหลักที่จะต้องทำการสุ่มหาตำแหน่งเพื่อที่จะเลื่อน สมมติให้ตัวแทนคำตอบตำแหน่งที่ 5 สุ่มไปเจอกับตัวแทนคำตอบตำแหน่งที่ 2 และนำเอาตัวแทนคำตอบตำแหน่งที่ 5 ไปแทนในช่องคำแทนคำตอบตำแหน่งที่ 2 จากนั้นนำตัวแทนคำตอบตำแหน่งที่ 2 แทนใน 3 ตัวแทนคำตอบตำแหน่งที่ 3 แทนใน 4 ตัวแทนคำตอบตำแหน่งที่ 4 แทนใน 5 เมื่อเลื่อนเรียบร้อยแล้วจะเป็นตัวแทนคำตอบใหม่ที่ได้ ดังรูปที่ 4.19



รูปที่ 4.19 การเลื่อนโดยยึดตำแหน่ง 5 เป็นตัวหลัก

4.3.2.7 จากการที่ได้ทำการสลับดังตัวอย่างข้างต้นจะเห็นได้ว่าจะมีตัวแทนคำตอบใหม่ที่ได้เกิดขึ้น ได้แก่

ก. 1 – 3 – 4 – 2 - 5

ข. 2 – 3 – 4 – 5 - 1

ค. 3 – 1 – 2 – 4 - 5

ง. 4 – 1 – 2 – 3 - 5

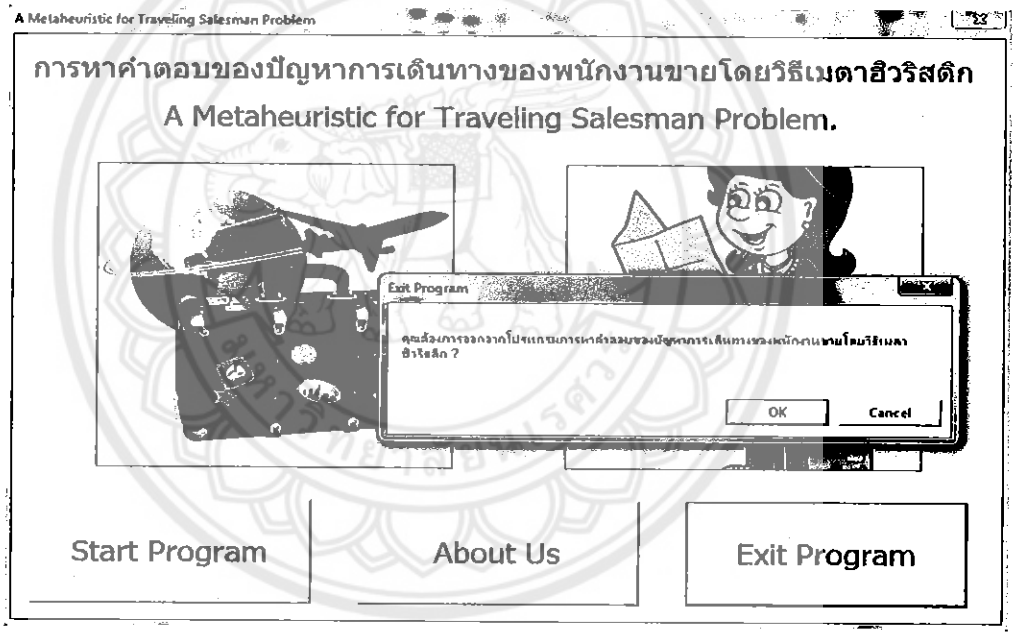
จ. 1 – 5 – 2 – 3 - 4

สรุป ในการสลับที่แต่ละครั้งจะมีตัวแทนคำตอบใหม่ที่ได้ครั้งละ 1 คำตอบ หรือถ้ามีเมืองจำนวน n เมือง ตัวแทนคำตอบใหม่ที่ได้จะมี n ตัวแทนคำตอบใหม่

4.4 ความสามารถของโปรแกรม และวิธีการใช้งานอย่างย่อ

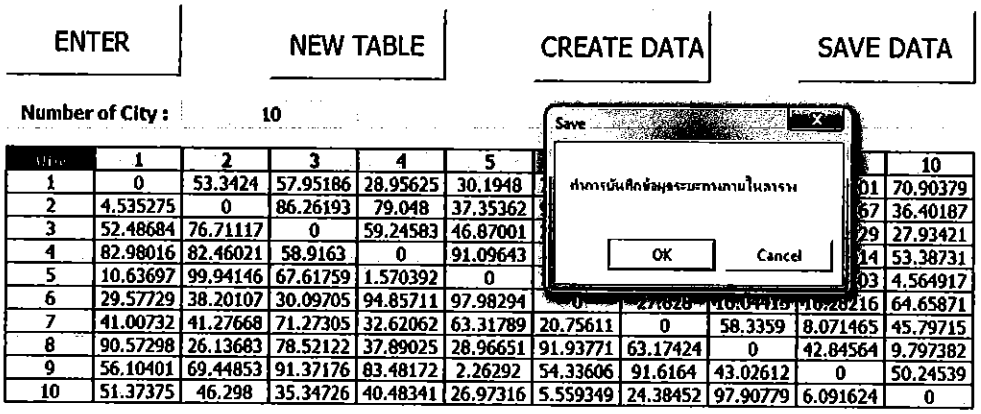
โปรแกรมการแก้ปัญหาการเดินทางของพนักงานขายโดยวิธีเมตาฮิวริสติกที่สร้างขึ้นเป็นเครื่องมือที่ช่วยประกอบการตัดสินใจในการวางแผนการขนส่งในโปรแกรมช่วยนี้ จะมีความสามารถแตกต่างกันไป จะขออธิบายความสามารถ และวิธีใช้แบบย่อดังนี้

4.4.1 เมื่อเปิดโปรแกรมช่วย จะมีหน้าต่างของโปรแกรมขึ้นมาแล้วกดที่ปุ่ม “Start Program” จะมีหน้าต่างขึ้นมาให้ผู้ใช้ใส่จำนวนเมืองที่ต้องการ และเมื่อผู้ใช้ต้องการออกจากโปรแกรมให้กดที่ปุ่ม “Exit Program” ของหน้าต่างโปรแกรม โดยก่อนจะปิดโปรแกรมจะมีหน้าต่างชื่อ Exit Program ขึ้นมาให้ผู้ใช้เห็นและมีข้อความว่า “คุณต้องการออกจากโปรแกรมการหาคำตอบของปัญหาการเดินทางของพนักงานขายโดยวิธีเมตาฮิวริสติก?” ถ้าผู้ใช้ต้องการออกจากโปรแกรมให้กดที่ปุ่ม “OK” ถ้าผู้ใช้ไม่ต้องการออกจากโปรแกรมให้กดที่ปุ่ม “Cancel” ดังรูปที่ 4.20



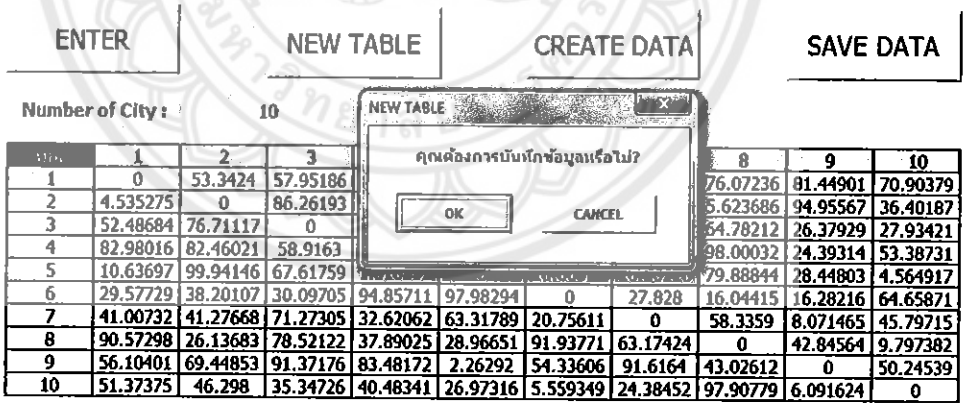
รูปที่ 4.20 Exit Program

4.4.2 เมื่อใส่จำนวนเมืองลงในช่อง “NUMBER OF CITY” แล้ว โปรแกรมจะมีตารางขึ้นมาให้ใส่ระยะทางจากเมืองหนึ่งไปยังอีกเมืองหนึ่ง เมื่อต้องการจะบันทึกข้อมูลระยะทางที่กรอกลงในให้ผู้ใช้กดที่ปุ่ม “SAVE DATA” จะมีหน้าต่างชื่อ Save ขึ้นมามีข้อความว่า “ทำการบันทึกข้อมูลภายในตาราง” ถ้าผู้ใช้จะบันทึกข้อมูลให้กดที่ปุ่ม “OK” ข้อมูลในตารางจะทำการบันทึกทันที ถ้าผู้ใช้ยังไม่บันทึกข้อมูลให้กดที่ปุ่ม “Cancel” ดังรูปที่ 4.21



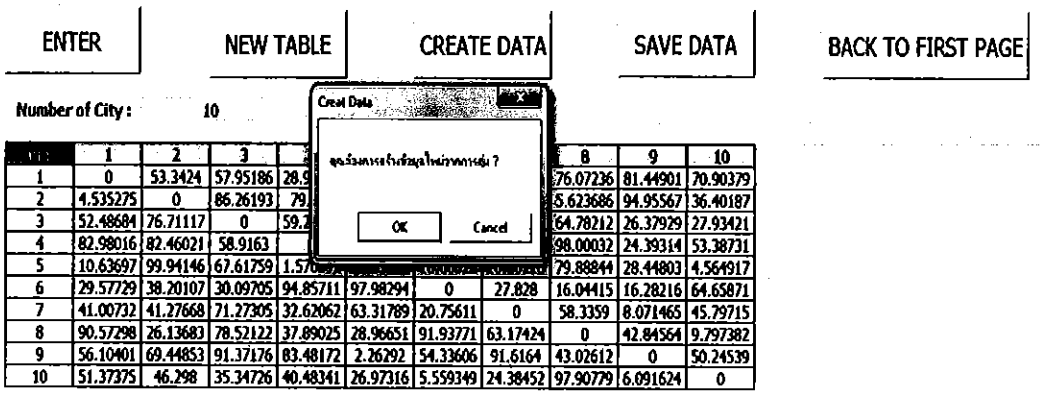
รูปที่ 4.21 Save

4.4.3 เมื่อผู้ใช้ต้องการที่จะทำตารางบันทึกระยะทางขึ้นใหม่ ให้ผู้ใช้กดที่ปุ่ม "NEW TABLE" จะมีหน้าต่างชื่อ NEW TABLE ขึ้นมา มีข้อความว่า "คุณต้องการบันทึกข้อมูลหรือไม่?" ถ้าต้องการบันทึกข้อมูลให้กดที่ปุ่ม "OK" ถ้าไม่ต้องการบันทึกข้อมูลให้กดที่ปุ่ม "CANCEL" ดังรูปที่ 4.22 เมื่อกดปุ่มที่ผู้ใช้ต้องการแล้ว โปรแกรมช่วยจะขึ้นหน้าต่างที่ชื่อ Input Number Of City หรือหน้าต่างให้ใส่จำนวนเมือง ให้ผู้ใช้ใส่จำนวนเมืองที่ต้องการสร้างลงไป



รูปที่ 4.22 NEW TABLE

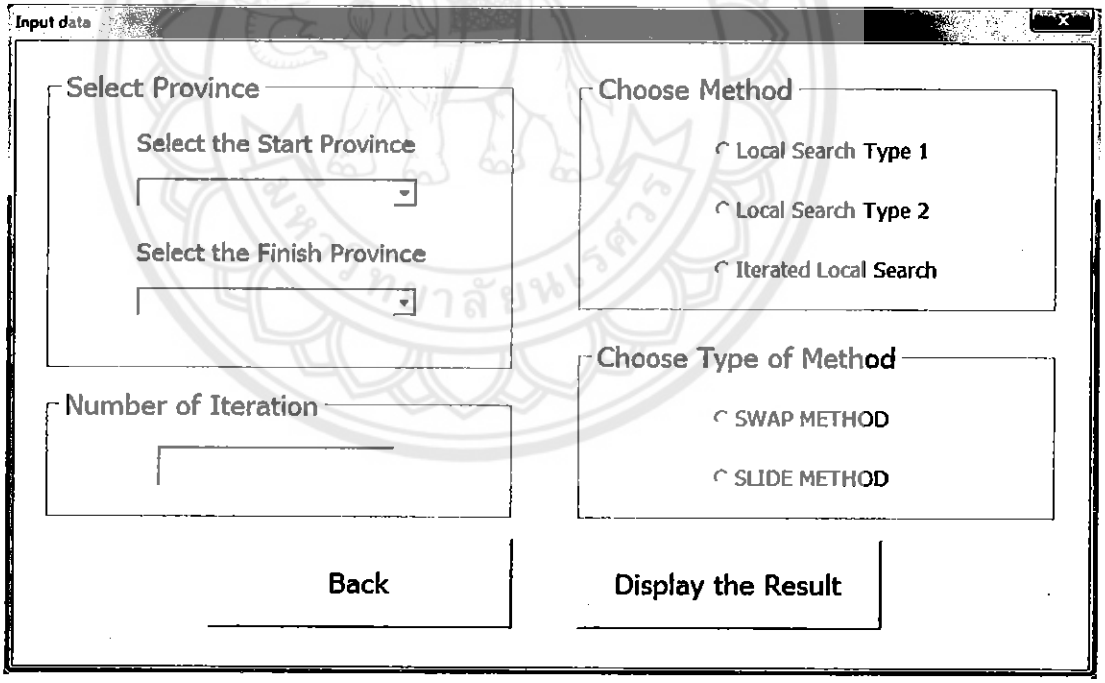
4.4.4 เมื่อผู้ใช้ต้องการที่จะสร้างข้อมูลระยะทางจากการสุ่มให้กดที่ปุ่ม "CREATE DATA" จะมีหน้าต่างชื่อ Create Data ขึ้นมา มีข้อความว่า "คุณต้องการสร้างข้อมูลใหม่จากการสุ่ม?" ถ้าผู้ใช้ต้องการสร้างข้อมูลใหม่จากการสุ่มให้กดที่ปุ่ม "OK" แต่ถ้าผู้ใช้ไม่ต้องการสร้างข้อมูลใหม่จากการสุ่มให้กดที่ปุ่ม "Cancel" ดังรูปที่ 4.23



	1	2	3					8	9	10
1	0	53.3424	57.95186	28.9				76.07236	81.44901	70.90379
2	4.535275	0	86.26193	79				5.623686	94.95567	36.40187
3	52.48684	76.71117	0	59.2				64.78212	26.37929	27.93421
4	82.98016	82.46021	58.9163					98.00032	24.39314	53.38731
5	10.63697	99.94146	67.61759	1.57				79.88844	28.44803	4.564917
6	29.57729	38.20107	30.09705	94.85711	97.98294	0	27.828	16.04415	16.28216	64.65871
7	41.00732	41.27668	71.27305	32.62062	63.31789	20.75611	0	58.3359	8.071465	45.79715
8	90.57298	26.13683	78.52122	37.89025	28.96651	91.93771	63.17424	0	42.84564	9.797382
9	56.10401	69.44853	91.37176	83.48172	2.26292	54.33606	91.6164	43.02612	0	50.24539
10	51.37375	46.298	35.34726	40.48341	26.97316	5.559349	24.38452	97.90779	6.091624	0

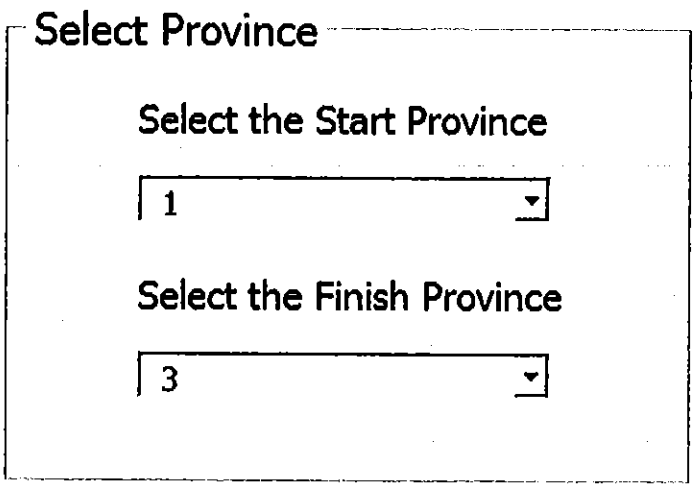
รูปที่ 4.23 Create Data

4.4.5 เมื่อใส่ระยะทางจากเมืองหนึ่งไปยังอีกเมืองหนึ่งเรียบร้อยแล้ว จะมีหน้าต่างชื่อ CHECK DISTANCE ขึ้นมา ให้กดที่ปุ่ม "OK" จากนั้นจะมีหน้าต่างชื่อ Input Data แสดงขึ้นมาให้ใส่ค่าตามหัวข้อต่างๆ ดังรูปที่ 4.24 ดังนี้



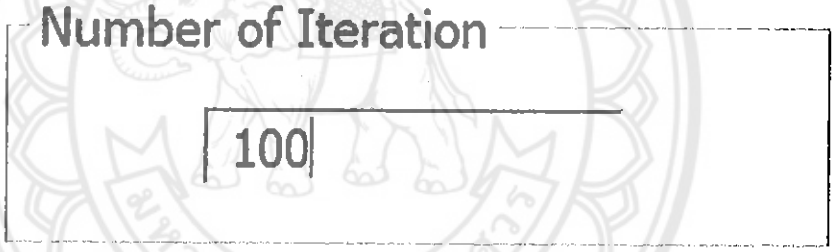
รูปที่ 4.24 Input Data

4.4.5.1 Select Province เป็นหัวข้อที่จะให้ผู้ใช้เลือกเมืองเริ่มต้นและเมืองสิ้นสุด โดยเมืองเริ่มต้นให้เลือกที่ "Select the Start Province" และเมืองสิ้นสุดให้เลือกที่ "Select the Finish Province" แต่ถ้าผู้ใช้ไม่ต้องการเลือกเมืองเริ่มต้นและเมืองสิ้นสุดให้เว้นเป็นช่องว่างไว้ ดังรูปที่ 4.25



รูปที่ 4.25 Select Province

4.4.5.2 Number of Iteration เป็นหัวข้อที่จะให้ผู้ใช้เลือกจำนวนของการวนรอบ โดยให้ผู้ใช้กรอกตัวเลขลงในช่องที่กำหนด ดังรูปที่ 4.26



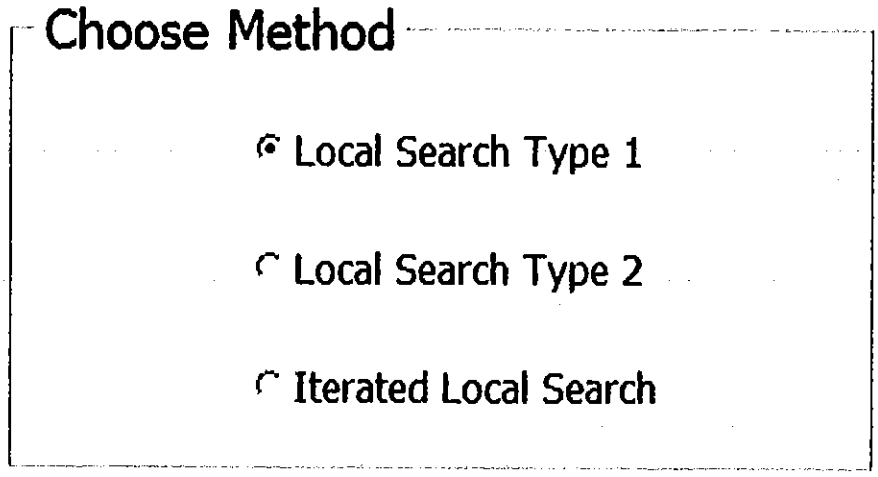
รูปที่ 4.26 Number of Iteration

4.4.5.3 Choose Method เป็นหัวข้อที่จะให้ผู้ใช้เลือกวิธีการในการหาคำตอบ ดังรูปที่ 4.27

ก. Local Search Type 1

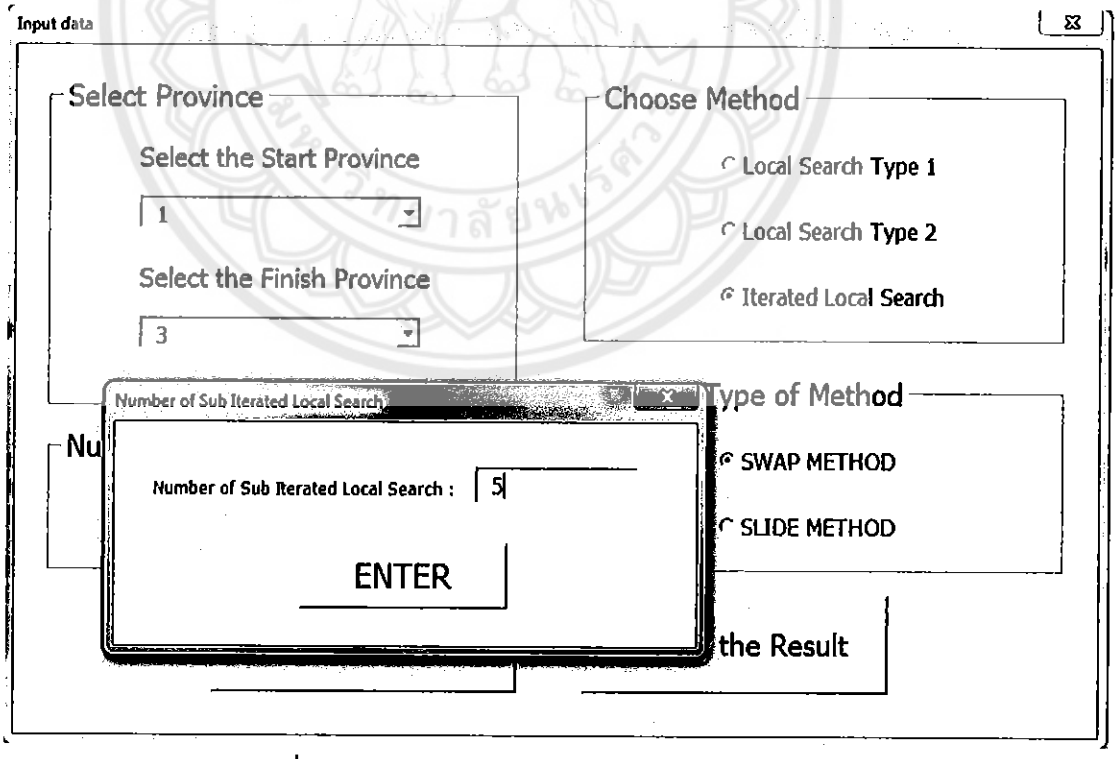
ข. Local Search Type 2

ค. Iterated Local Search



รูปที่ 4.27 Choose Method

เมื่อผู้ใช้เลือกวิธีการในการหาคำตอบโดยใช้ Iterated Local Search โปรแกรมจะมีหน้าต่าง ชื่อว่า Number of Sub Iterated Local Search ให้ผู้ใช้ใส่จำนวนของการวนเพื่อที่จะเมื่อไม่พบคำตอบที่ดีกว่าเดิมตามจำนวนรอบของการวนที่กำหนดให้สุ่มหาชุดคำตอบใหม่ ดังรูปที่ 4.28

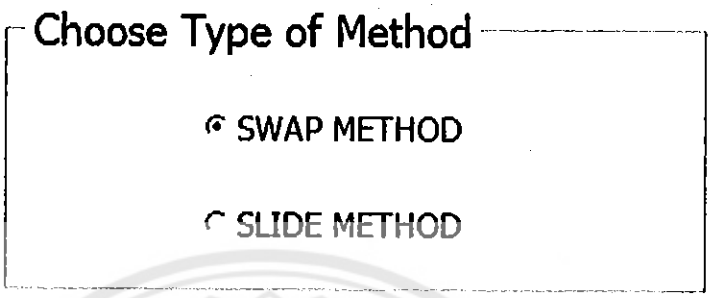


รูปที่ 4.28 Number of Sub Iterated Local Search

4.4.5.4 Choose Type of Method เป็นหัวข้อที่จะให้ผู้ใช้เลือกวิธีการปรับปรุงคำตอบ ดังรูปที่ 4.29

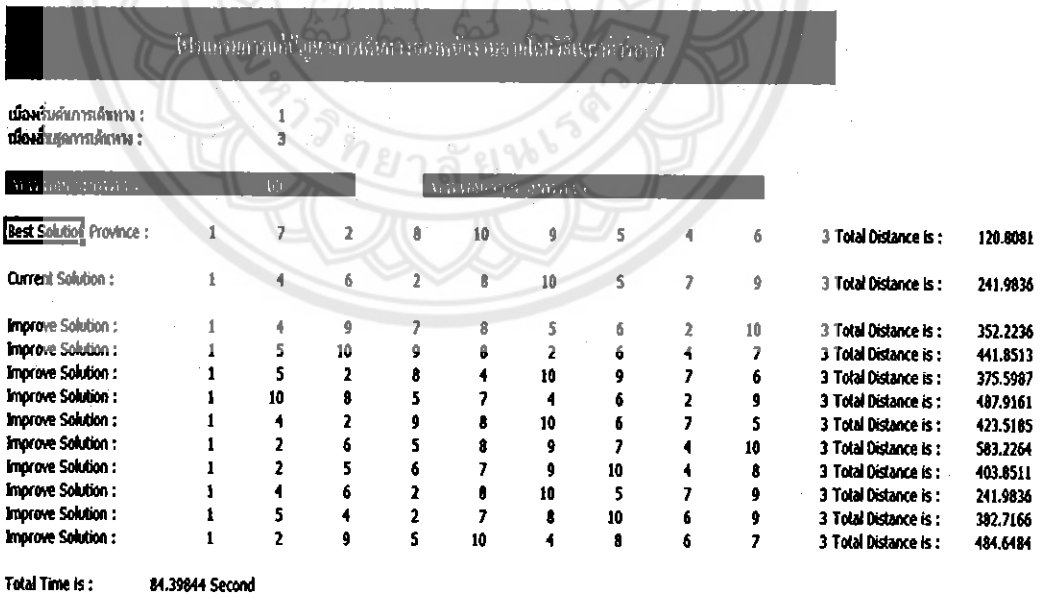
ก. Swap Method

ข. Slide Method



รูปที่ 4.29 Choose Type of Method

4.4.6 เมื่อกรอกจำนวนของการวนรอบซ้ำและเลือกหัวข้อต่างๆ ที่ต้องการเรียนร้อยแล้ว ให้กดที่ปุ่ม "Display the Result" ตามรูปที่ 4.24 โปรแกรมจะทำการแก้ปัญหาการเดินทางของพนักงานขาย ดังรูปที่ 4.30



Best Solution Province :	1	7	2	8	10	9	5	4	6	3 Total Distance is :	120.8081
Current Solution :	1	4	6	2	8	10	5	7	9	3 Total Distance is :	241.9836
Improve Solution :	1	4	9	7	8	5	6	2	10	3 Total Distance is :	352.2236
Improve Solution :	1	5	10	9	8	2	6	4	7	3 Total Distance is :	441.8513
Improve Solution :	1	5	2	8	4	10	9	7	6	3 Total Distance is :	375.5987
Improve Solution :	1	10	8	5	7	4	6	2	9	3 Total Distance is :	487.9161
Improve Solution :	1	4	2	9	8	10	6	7	5	3 Total Distance is :	423.5185
Improve Solution :	1	2	6	5	8	9	7	4	10	3 Total Distance is :	583.2264
Improve Solution :	1	2	5	6	7	9	10	4	8	3 Total Distance is :	403.8511
Improve Solution :	1	4	6	2	8	10	5	7	9	3 Total Distance is :	241.9836
Improve Solution :	1	5	4	2	7	8	10	6	9	3 Total Distance is :	382.7166
Improve Solution :	1	2	9	5	10	4	8	6	7	3 Total Distance is :	484.6484

Total Time is : 84.39844 Second

รูปที่ 4.30 โปรแกรมจะทำการแก้ปัญหาการเดินทางของพนักงานขาย

4.5 การออกแบบ และผลการทดลองเพื่อเปรียบเทียบประสิทธิภาพของวิธีการหาคำตอบ

ในการเปรียบเทียบประสิทธิภาพการหาคำตอบของวิธีการหาคำตอบทั้ง 3 วิธี ในวิจัยได้มีการสร้างโจทย์ที่มีระยะทางจากการสุ่มมาทำการทดลองปัญหาขนาดละ 2 โจทย์ นั้นคือ ปัญหาขนาดเล็ก เมือง 2 โจทย์ ซึ่งปัญหาขนาดเล็กเป็นปัญหาที่มีจำนวนเมือง 10 เมือง ต่อมาจะเป็นปัญหาขนาดกลาง 2 โจทย์ มีจำนวนเมือง 25 เมือง และปัญหาขนาดใหญ่ 2 โจทย์ จะมีจำนวนเมือง 50 เมือง มาทำการทดสอบวิธีการหาคำตอบทั้ง 3 วิธีที่สร้างขึ้น โดยในการทดลองจะกำหนดให้วิธีการหาคำตอบทั้ง 3 มีการวนรอบหาคำตอบ 1000 รอบ แต่การหาคำตอบโดย Iterated Local Search จะกำหนดให้มีจำนวนของการวนรอบซ้ำ 50 รอบ ซึ่งมีผลการทดลองดังรูปที่ 4.31 รูปที่ 4.32 และรูปที่ 4.33 ตามลำดับ

		Local Search 1		Local Search 2		Iterated Local Search	
		Swap	Slide	Swap	Slide	Swap	Slide
10 เมือง โจทย์ 1	Distance	173.0126	172.572	226.1813	204.5606	179.4289	172.572
	SD	0.92882	0	19.29558	27.58506	13.51433	0
	Time	13.30547	13.44922	12.81953	12.87344	11.78438	11.79141
10 เมือง โจทย์ 2	Distance	694.7906	692.1275	756.0273	724.0159	704.8588	698.0898
	SD	6.431919	4.210681	38.37752	30.0779	10.43481	7.91877
	Time	13.41953	13.43594	12.86953	12.85469	11.75469	11.77422

รูปที่ 4.31 การทดลองปัญหาขนาดเล็ก

จากรูปที่ 4.31 แสดงผลของการหาคำตอบส่วนเบี่ยงเบนมาตรฐาน และเวลาจากโปรแกรม โดยระยะทางระหว่างเมือง 10 เมือง โจทย์ที่ 1 ได้จากการสุ่มตัวเลขขึ้นมาตั้งแต่ 1 ถึง 100 และโจทย์ที่ 2 ได้จากการสุ่มตัวเลขขึ้นมาตั้งแต่ 50 ถึง 200 ซึ่งจากการทดลองเห็นได้ว่า โจทย์ที่ 1 การค้นหาคำตอบแบบ Local Search Type 1 และ Iterated Local Search โดยวิธีการเสื่อนจะได้คำตอบของระยะทางที่ดีกว่าแบบสลับ โจทย์ที่ 2 การค้นหาคำตอบแบบ Local Search Type 1 โดยวิธีการเลื่อนจะได้คำตอบของระยะทางที่ดีกว่าแบบสลับ และได้คำตอบดีกว่า Local Search Type 2 กับ Iterated Local Search

		Local Search 1		Local Search 2		Iterated Local Search	
		Swap	Slide	Swap	Slide	Swap	Slide
25 เมือง โจทย์ 1	Distance	272.696	218.166	325.9768	278.2334	272.04	240.4418
	SD	17.03439	16.07793	33.33518	46.19913	29.16581	21.06628
	Time	76.1875	76.36846	75.07656	75.07666	72.41221	72.3623
25 เมือง โจทย์ 2	Distance	1691.194	1604.758	1825.27	1740.962	1732.194	1684.487
	SD	19.9306	27.66003	62.41555	64.2839	39.47174	27.67474
	Time	76.37002	76.3918	75.35283	72.90859	72.3748	71.90664

รูปที่ 4.32 การทดลองปัญหาขนาดกลาง

จากรูปที่ 4.32 แสดงผลของการหาคำตอบส่วนเบี่ยงเบนมาตรฐาน และเวลาจากโปรแกรม โดยระยะทางระหว่างเมือง 25 เมือง โจทย์ที่ 1 ได้จากการสุ่มตัวเลขขึ้นมาตั้งแต่ 1 ถึง 100 และโจทย์ที่ 2 ได้จากการสุ่มตัวเลขขึ้นมาตั้งแต่ 50 ถึง 200 ซึ่งจากการทดลองเห็นได้ว่า โจทย์ที่ 1 การค้นหาคำตอบโดยใช้ Local Search Type 1 แบบเลื่อน ได้คำตอบดีกว่า Local Search Type 2 (แบบเลื่อน และสลับ) กับ Iterated Local Search (แบบเลื่อน และสลับ) โจทย์ที่ 2 การค้นหาคำตอบโดยใช้ Local Search Type 1 แบบเลื่อน ได้คำตอบดีกว่า Local Search Type 2 (แบบเลื่อน และสลับ) กับ Iterated Local Search (แบบเลื่อน และสลับ)

		Local Search 1		Local Search 2		Iterated Local Search	
		Swap	Slide	Swap	Slide	Swap	Slide
50 เมือง โจทย์ 1	Distance	540.7917	431.7335	617.0261	483.4656	624.7115	572.6012
	SD	17.49752	27.08478	44.53739	72.55469	40.94434	19.70645
	Time	296.1141	309.091	299.2594	304.2008	326.6383	292.5023
50 เมือง โจทย์ 2	Distance	3233.442	3079.937	3344.381	3152.947	3386.337	3309.683
	SD	51.2496	49.72263	80.43	92.3108	40.10301	39.66208
	Time	330.0094	295.4642	307.4727	294.6322	376.5211	296.4511

รูปที่ 4.33 การทดลองปัญหาขนาดใหญ่

จากรูปที่ 4.33 แสดงผลของการหาคำตอบส่วนเบี่ยงเบนมาตรฐาน และเวลาจากโปรแกรม โดยระยะทางระหว่างเมือง 50 เมือง โจทย์ที่ 1 ได้จากการสุ่มตัวเลขขึ้นมาตั้งแต่ 1 ถึง 100 และโจทย์ที่ 1 ได้จากการสุ่มตัวเลขขึ้นมาตั้งแต่ 50 ถึง 200 ซึ่งจากการทดลองเห็นได้ว่า โจทย์ที่ 1 การค้นหาคำตอบโดยใช้ Local Search Type 1 แบบเลื่อน ได้คำตอบดีกว่า Local Search Type 2 (แบบเลื่อน และสลับ) กับ Iterated Local Search (แบบเลื่อน และสลับ) โจทย์ที่ 2 การค้นหาคำตอบโดยใช้ Local Search Type 1 แบบเลื่อน ได้คำตอบดีกว่า Local Search Type 2 (แบบเลื่อน และสลับ) กับ Iterated Local Search (แบบเลื่อน และสลับ)

4.6 การวิเคราะห์ความแปรปรวน (ANOVA) ของผลลัพธ์โดยใช้โปรแกรม Minitab 16

การวิเคราะห์ความแปรปรวนเป็นหนึ่งในวิธีการทางสถิติที่นิยมใช้กันอย่างทั่วไปในการตัดสินใจทางสถิติคือการทดสอบสมมติฐาน (Hypothesis Testing) ซึ่งในโปรแกรม Minitab 16 นั้นจะมีคำสั่งในเรื่องการทดสอบสมมติฐานมากมาย รวมถึงการวิเคราะห์ความแปรปรวน โดยทั่วไปการทดสอบสมมติฐานจะสันนิษฐานว่าสิ่งที่เราสนใจนั้นเป็นจริงก่อน แล้วจึงทำการพิสูจน์ด้วยข้อมูลจากกลุ่มตัวอย่าง

การทดสอบสมมติฐานประกอบด้วย 2 ส่วน คือ สมมติฐานหลัก (เขียนแทนด้วย H_0) และสมมติฐานรอง (เขียนแทนด้วย H_1) สมมติฐานหลักเป็นข้อสันนิษฐานเบื้องต้น ซึ่งมักอ้างอิงจากผลการศึกษาก่อนหน้านี้หรือความรู้ทั่วไป ส่วนสมมติฐานรองคือสิ่งที่ต้องการพิสูจน์ว่าเป็นจริง

ซึ่งการวิเคราะห์ความแปรปรวนนี้ จะเป็นการวิเคราะห์ว่า มีความแตกต่างระหว่างค่าเฉลี่ยของค่าใช้จ่าย หรือระยะทางในการเดินทางของปัญหาการเดินทางของพนักงานขาย ซึ่งเป็นค่าของผลลัพธ์และวิธีการปรับปรุงคำตอบทั้งแบบสลับ และแบบเลื่อน ซึ่งเป็นปัจจัยที่มีผลกระทบต่อค่าของผลลัพธ์อย่างมีนัยสำคัญทางสถิติหรือไม่ สามารถตรวจสอบได้โดยการวิเคราะห์ความแปรปรวนแบบปัจจัยเดียว (One – Way ANOVA) โดยเป็นการทดสอบว่าค่าเฉลี่ยสำหรับข้อมูลสองกลุ่มขึ้นไป ที่มีการแบ่งกลุ่มโดยปัจจัยเดียวนั้นเท่ากันหรือไม่

4.6.1 การแปรผลจากโปรแกรม Minitab 16 จากการวิเคราะห์ความแปรปรวนระหว่างค่าเฉลี่ยของค่าใช้จ่าย หรือระยะทางในการเดินทางในแต่ละวิธีการ ของการแก้ปัญหาการเดินทางของพนักงานขายโดยวิธีเมตาฮิวริสติก

กระบวนการตัดสินใจสำหรับการทดสอบสมมติฐานนั้นอยู่บนพื้นฐานของความน่าจะเป็นทางสถิติหรือเราเรียกว่าค่า P – Value ที่ได้จากการทดสอบ

4.6.1.1 ถ้าค่า P – Value มีค่าน้อยกว่าหรือเท่ากับระดับนัยสำคัญที่ตั้งไว้ (α - Level) เท่ากับ 0.05 ให้ปฏิเสธสมมติฐานหลัก (H_0) ซึ่งก็คือ วิธีการปรับปรุงคำตอบทั้งแบบสลับและแบบเลื่อนนั้น ไม่มีผลกระทบต่อการหาค่าคำตอบ และยอมรับสมมติฐานรอง (H_1) ซึ่งก็คือ วิธีการปรับปรุงคำตอบแบบสลับหรือแบบเลื่อนนั้น มีผลต่อการหาค่าคำตอบ

4.6.1.2 ถ้าค่า P – Value มีค่ามากกว่าระดับนัยสำคัญที่ตั้งไว้ (α - Level) เท่ากับ 0.05 แสดงว่าไม่สามารถปฏิเสธสมมติฐานหลักได้ เท่ากับว่าวิธีการปรับปรุงคำตอบทั้งแบบสลับและแบบเลื่อนนั้น ไม่มีผลกระทบต่อการหาค่าคำตอบ

4.6.2 สรุปผลการวิเคราะห์ความแปรปรวนระหว่างค่าเฉลี่ยของค่าใช้จ่าย หรือระยะทางในการเดินทางในแต่ละวิธีการ ของการแก้ปัญหาการเดินทางของพนักงานขายโดยวิธีเมตาฮิวริสติก

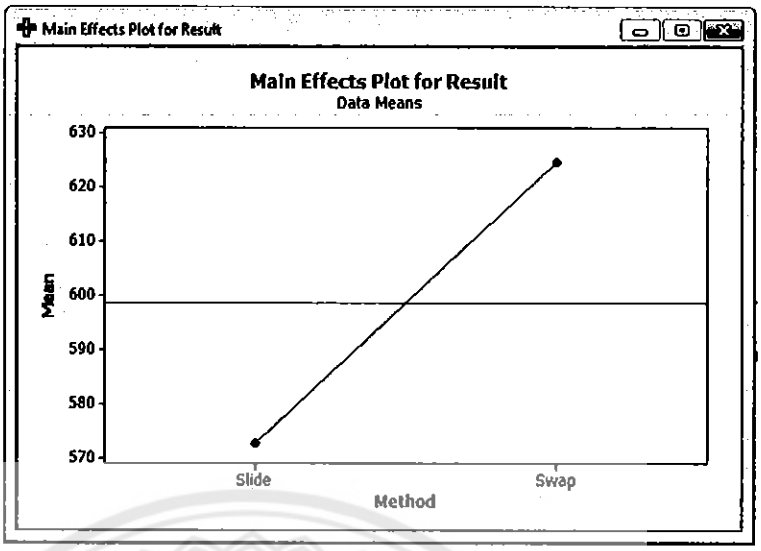
ค่าความน่าจะเป็นทางสถิติของโจทย์ปัญหาการเดินทางของพนักงานขายขนาด 10 เมือง 25 เมือง และ 50 เมือง แสดงไว้ดังตารางที่ 1 และกำหนดให้ระดับนัยสำคัญ มีค่าเท่ากับ 0.05

ตารางที่ 4.1 สรุปผลการวิเคราะห์ความแปรปรวน

รายละเอียด	P - Value		
	Local Search 1	Local Search 2	Iterated Local Search
	Swap – Slide	Swap - Slide	Swap - Slide
4.6.2.1 โจทย์ขนาด 10 เมือง โจทย์ที่ 1	0.151	0.057	0.126
4.6.2.2 โจทย์ขนาด 10 เมือง โจทย์ที่ 2	0.288	0.052	0.120
4.6.2.3 โจทย์ขนาด 25 เมือง โจทย์ที่ 1	0.000	0.016	0.012
4.6.2.4 โจทย์ขนาด 25 เมือง โจทย์ที่ 2	0.000	0.008	0.006
4.6.2.5 โจทย์ขนาด 50 เมือง โจทย์ที่ 1	0.000	0.000	0.002
4.6.2.6 โจทย์ขนาด 50 เมือง โจทย์ที่ 2	0.000	0.000	0.000

จากตารางที่ 4.1 สามารถสรุปได้ว่าค่า P – Value ของโจทย์ขนาด 10 เมืองทั้งโจทย์ที่ 1 และโจทย์ที่ 2 มีค่ามากกว่าระดับนัยสำคัญที่ตั้งไว้เท่ากับ 0.05 แสดงว่าไม่สามารถปฏิเสธสมมติฐานหลักซึ่งก็คือ วิธีการปรับปรุงคำตอบทั้งแบบสลับ และแบบเลื่อนที่ใช้นั้น ไม่มีผลกระทบต่อการหาค่าคำตอบได้ จึงเป็นเหตุให้ไม่สามารถยอมรับสมมติฐานรอง ที่วิธีการปรับปรุงคำตอบ มีผลต่อการหาคำตอบตามไปด้วย

ในส่วนของโจทย์ขนาด 25 เมือง และ 50 เมือง ทั้งโจทย์ที่ 1 และโจทย์ที่ 2 แสดงให้เห็นว่าค่า P – Value มีค่าน้อยกว่าระดับนัยสำคัญที่ตั้งไว้เท่ากับ 0.05 แสดงว่าสามารถปฏิเสธสมมติฐานหลัก และยอมรับสมมติฐานรอง ซึ่งก็คือ วิธีการปรับปรุงคำตอบทั้งแบบสลับ และแบบเลื่อนที่ใช้นั้น มีผลต่อการหาค่าคำตอบ โดยวิธีการที่สามารถหาค่าคำตอบได้ดีว่าของโจทย์ขนาด 25 เมืองและ 50 เมือง ทั้งโจทย์ที่ 1 และโจทย์ที่ 2 คือ วิธีการปรับปรุงคำตอบแบบเลื่อนที่สามารถหาระยะทางรวมหรือค่าใช้จ่ายได้น้อยกว่าแบบสลับ ซึ่งสามารถแสดงตัวอย่างผลการวิเคราะห์รูปแบบของการหาคำตอบแบบสลับ และแบบเลื่อน ของโจทย์ขนาด 50 เมือง โจทย์ที่ 1 วิธีการหาคำตอบแบบ Iterated Local Search แสดงดังรูปที่ 4.34



รูปที่ 4.34 ตัวอย่างผลการวิเคราะห์รูปแบบของวิธีการปรับปรุงคำตอบแบบสลับและแบบเลื่อน

บทที่ 5
บทสรุปและข้อเสนอแนะ

ในส่วนนี้จะกล่าวถึงการสรุปผลจากการดำเนินโครงงาน รวมถึงปัญหาที่พบในระหว่างดำเนินโครงงาน และแนวทางในการแก้ปัญหา ดังข้อที่ 5.1 – 5.3

5.1 สรุปผลการดำเนินโครงงาน

จากการดำเนินโครงงาน การหาคำตอบของปัญหาการเดินทางของพนักงานขายโดยวิธีเมตาฮิว - ริสติก สามารถสรุปผลการดำเนินโครงงานได้ดังนี้

ในการดำเนินโครงงานผู้ดำเนินโครงงานได้ทำการออกแบบและสร้างโปรแกรมที่แก้ปัญหาของปัญหาการเดินทางของพนักงานขาย โดยมีเป้าหมายเพื่อจัดลำดับเส้นทางการเดินทาง การหาเส้นทางที่เหมาะสมที่สุดสำหรับการเดินทาง เพื่อที่จะทำให้ประหยัดเวลา และลดค่าใช้จ่ายในการเดินทาง ซึ่งผู้ดำเนินโครงงานได้ออกแบบเมตาฮิวริสติกหรือการหาคำตอบเพื่อที่จะใช้เขียนโปรแกรมของการแก้ปัญหาการเดินทางของพนักงานขายมา 3 แบบ คือ

5.1.1 Local Search Type 1

5.1.2 Local Search Type 2

5.1.3 Iterated Local Search

จากนั้นผู้ดำเนินโครงงานได้ทำการเขียนโปรแกรมเพื่อแก้ปัญหาการเดินทางของพนักงานขายโดยใช้ Visual Basic for Applications (VBA) บน Microsoft Excel ซึ่งโปรแกรมที่ได้ จะเป็นโปรแกรมที่มีการหาระยะทางการเดินทางของพนักงานขาย ที่ไปยังเมืองต่างๆ ให้มีระยะทางที่น้อยที่สุด

จากการที่ได้มีการทดลองใช้โปรแกรมการหาคำตอบของปัญหาการเดินทางของพนักงานขาย สรุปได้ว่าในการหาค่าคำตอบของปัญหาการเดินทางของพนักงานขาย จากโจทย์ที่มีการกำหนดขึ้นมา เห็นได้ว่าวิธีการในการหาคำตอบ Local Search Type 1 โดยใช้วิธีการปรับปรุงคำตอบแบบเลื่อน มีคำตอบที่ดีกว่าวิธีการในการหาคำตอบ Local Search Type 2 และวิธีการในการหาคำตอบ Iterated Local Search

5.2 ปัญหาที่พบระหว่างการดำเนินโครงงาน

5.2.1 ขั้นตอนในการออกแบบเมตาฮิวริสติก ยังมีความรู้เกี่ยวกับเมตาฮิวริสติกน้อย ทำให้ใช้เวลาในการดำเนินงานของส่วนนี้พอสมควร

5.2.2 ขั้นการสร้างโปรแกรมโดยใช้ Visual Basic Application (VBA) ใช้เวลาในการเขียนโปรแกรมนาน เนื่องจากไม่มีความรู้ในโค้ดพื้นฐานของ VBA มาก่อน จึงต้องศึกษาค้นคว้าด้วยตนเอง และมีการแก้ไขหลายครั้งหลังจากที่นำไปทดลอง

5.3 แนวทางในการแก้ปัญหา

5.3.1 ขอคำชี้แนะจากอาจารย์ที่ปรึกษา และค้นคว้าจากผลงานวิจัยต่างๆ รวบถึงสืบค้นข้อมูลต่างๆ จากทางอินเตอร์เน็ต

5.3.2 สอบถามจากผู้มีความรู้ในการเขียนโปรแกรม VBA ค้นคว้าจากผลงานวิจัยต่างๆ ที่เกี่ยวข้อง และศึกษาจากหนังสือที่เขียนเกี่ยวกับ VBA รวบถึงการสืบค้นข้อมูลต่างๆ ทางอินเตอร์เน็ต

เอกสารอ้างอิง

กมลทิพย์ ขานทอง. (2544). **การแก้ปัญหาการเดินทางเซลล์แมนแบบคอขวดโดยวิธีดัดแปลงของอิสต์แมน.** วิทยานิพนธ์ วศ.ม., มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ธีรพล อร่ามเกียรติสิริ. (2547). **ขั้นตอนวิธีลงตัวสำหรับปัญหาการเดินทางของพนักงานขายแบบเกาะกลุ่ม.** วิทยานิพนธ์ ดุษฎีบัณฑิต., มหาวิทยาลัยเกษตรศาสตร์

มาลินี หลวงคลัง และยุภาภัทร อนิลบล. (2554). **โปรแกรมช่วยจัดเส้นทางโดยใช้วิธีเจนเนติกอัลกอริทึม.** ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหการ มหาวิทยาลัยนเรศวร

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

หทัยทิพย์ ภุชงควาริน. (2547). **ปัญหาการเดินทางของพนักงานขายที่มีระยะเวลาไม่แน่นอนแบบคงตัว.** วิทยานิพนธ์ วศ.ม., มหาวิทยาลัยเกษตรศาสตร์

David J. Rader. (2010). **Deterministic Operations Research: Models and Methods in Linear Optimization.** English : Wiley

ประวัติผู้ดำเนินโครงงาน

ชื่อ นายสายสัมพันธ์ ซุ้นเจริญ

ภูมิลำเนา 39 หมู่ 17 ต.แม่นาเรือ อ.เมือง จ.พะเยา

ประวัติการศึกษา

- จบจากระดับมัธยมจากโรงเรียนพะเยาพิทยาคม จ.พะเยา
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : L_inum@hotmail.com

ชื่อ นายอธิวัฒน์ กุลศิริ

ภูมิลำเนา 821 ต.ศรีพนมมาศ อ.ลับแล จ.อุตรดิตถ์

ประวัติการศึกษา

- จบจากระดับมัธยมจากโรงเรียนลับแลพิทยาคม จ.อุตรดิตถ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : Atiwat_12@hotmail.com