รายงานวิจัย

เรื่อง

การจัดเส้นทางการขนส่งด้วยวิธีเมต้าฮิวริสติกส์ กรณีศึกษา บริษัทผู้ให้บริการขนส่งทางด้านโลจิสติกส์

Metaheuristic for Transportation Routing: A Case Study of Transportation Company

ณัตพร ไชยเสนา

Nattaporn Chaisena

การวิจัยครั้งนี้ได้รับเงินทุนการวิจัยจากมหาวิทยาลัยราชพฤกษ์

ปีการศึกษา 2560

ชื่องานวิจัย: การจัดเส้นทางการขนส่งด้วยวิธีเมต้าฮิวริสติกส์: กรณีศึกษา บริษัทผู้ให้บริการขนส่ง

ชื่อผู้วิจัย: ณัตพร ไชยเสนา

ปีที่ทำงานวิจัยแล้วเสร็จ: 2560

บทคัดย่อ

การวิจัยครั้งนี้มีจุดประสงค์เพื่อนำเทคนิคในการจัดเส้นทางการขนส่งด้วยวิธีการเมต้าฮิว-ริสติกส์มาใช้ในการจัดเส้นทางการขนส่งสินค้าให้กับบริษัทกรณีศึกษาและนำมาช่วยในการลดระยะทางในการขนส่งให้ได้ระยะทางที่สั้นที่สุด ผู้วิจัยได้มีการออกแบบจำลองและปรับปรุงให้ใกล้เคียงสอดคล้องกับปัญหาจริง โดยนำเทคนิควิธีการเมต้าฮิวริสติกส์ด้วยวิธีการสลับจุดส่งทีละ 3 จุด (3-OPT)

ผลการวิจัย พบว่า การออกแบบจำลองการจัดเส้นทางเดินรถขนส่งสินค้าโดยได้นำหลักการเมต้าฮิวริสติกส์ด้วยวิธีการสลับจุด 3 จุด 3-OPT มาประยุกต์กับโปรแกรมจัดเส้นทางที่เขียนชุดคำสั่งลงบนโปรแกรม Visual Basic for Application (VBA) และทำงานบนโปรแกรม Microsoft Excel และดึงข้อมูลระยะทางด้วยระบบสารสนเทศทางภูมิศาสตร์ โดยการเก็บข้อมูลตัวอย่าง 7 ชุดข้อมูล ระหว่างวันที่ 6 -12 พฤศจิกายน 2560 ทดลองจัดเส้นทางด้วยวิธี 3-OPT โดยเลือกรถบรรทุกประเภท 6 ล้อกลาง ผลที่ได้จากการทดลองสามารถลดระยะทางขนส่งรวมจากเดิม 5,005.93 กิโลเมตร เหลือเพียง 4,512.11 กิโลเมตร โดยสามารถลดระยะทางจากเดิมลง 493.82 กิโลเมตร และยังสามารถลดต้นทุนค่าน้ำมันเชื้อเพลิงลงได้ 7 วัน 12,389.94 บาท

คำสำคัญ: การจัดเส้นทางเดินรถขนส่งสินค้า วิธีการเมต้าฮิวริสติกส์ วิธีการสลับจุดส่งทีละ 3 จุด (3-OPT)

Research Title: Metaheuristic for Transportation Routing: A Case Study of Transportation Company

Researcher: Nattaporn Chaisena

Year: 2017

Abstract

The research purposes were to manage and reduce transportation routing of transportation company using metaheuristic technique. Design model corresponds to problem by 3-OPT. Applied the 3-OPT with routing software in Visual Basic for Application (VBA) and process on Microsoft Excel. The samples were 7 routes distance from Geographic Information System (GIS) of six-wheel truck between November 6 – 12, 2017.

The results found that;

The vehicle routing reduced the transportation distance from 5,005.93 KM to 4,512.11 KM; the distance reduced for 493.82 KM and reduced the cost of fuel for 12,389.94 Baht in 7 days.

Keywords: transportation routing, metaheuristics, 3-OPT

กิตติกรรมประกาศ

งานวิจัยเรื่อง การจัดเส้นทางการขนส่งด้วยวิธีเมต้าฮิวริสติกส์ กรณีศึกษา บริษัทผู้ให้บริการขนส่ง เสร็จสมบูรณ์ได้ด้วยความกรุณาจากมหาวิทยาลัยราชพฤกษ์ ดร.อณาวุฒิ ชูทรัพย์ อธิการบดี ประธานกรรมการวิจัย คณะกรรมการวิจัยทุกท่านที่ได้พิจารณาทุนอุดหนุนการวิจัยให้แก่ข้าพเจ้า รวมถึงที่ปรึกษาโครงการวิจัยดร.สุภาภรณ์ ตั้งดำเนินสวัสดิ์ ที่ได้ให้คำแนะนำและช่วยแก้ไขข้อบกพร่องมาตลอด ซึ่งทำให้งานวิจัยเรื่องนี้เสร็จสมบูรณ์

มกราคม 2561

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญภาพ ..... ช

บทที่ 1 บทนำ 1

ความเป็นมาและความสำคัญของปัญหา 1

คำถามการวิจัย 2

วัตถุประสงค์ของการวิจัย 2

ขอบเขตการวิจัย 2

ประโยชน์ที่คาดว่าจะได้รับ 2

นิยามคำศัพท์เฉพาะ 3

บทที่ 2 แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง 4

ความหมายของการขนส่ง 4

รูปแบบของการขนส่งสินค้า 5

ลักษณะของการขนส่งสินค้าด้วยรถบรรทุก 6

การจัดเส้นทางการเดินรถ (Vehicle Routing) 7

ปัญหาการจัดเส้นทางการเดินรถ 9

เทคนิคการหาคำตอบ 14

ระบบสารสนเทศทางภูมิศาสตร์ 23

Visual Basic 29

งานวิจัยที่เกี่ยวข้อง 29

บทที่ 3 ระเบียบวิธีวิจัย 34

ประชากรที่ใช้ในการศึกษา 34

ขั้นตอนการดำเนินงานวิจัย 34

เครื่องมือที่ใช้ในงานวิจัย 35

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการวิจัย 37
การออกแบบการจัดเส้นทางเดินรถขนส่งสินค้า..... 37
แบบจำลองทางคณิตศาสตร์..... 40
ต้นทุนน้ำมันเชื้อเพลิง..... 41
ผลการทดลองการจัดเส้นทางเดินรถขนส่งสินค้า 41
บทที่ 5 อภิปรายผล และข้อเสนอแนะ 44
สรุปผลการวิจัย..... 44
อภิปรายผลการวิจัย 45
ข้อเสนอแนะ 46
ข้อเสนอแนะในการนำผลการวิจัยไปใช้..... 46
ข้อเสนอแนะในการนำทำวิจัยครั้งต่อไป..... 46
บรรณานุกรม..... 47
ประวัติผู้วิจัย..... 134

มหาวิทยาลัยราชภัฏ RAJAPRUK UNIVERSITY

สารบัญตาราง

ตารางที่ **หน้า**

2.1 เปรียบเทียบคุณลักษณะของรูปแบบการขนส่งต่าง ๆ 6
3.1 ประเภทและข้อจำกัดด้านน้ำหนักและปริมาตร..... 35
4.1 ตัวอย่างเมตริกซ์ระยะทาง 38
4.2 เปรียบเทียบระยะทางหลังจากการปรับปรุงเส้นทาง 43

สารบัญภาพ

ภาพที่ **หน้า**

2.1 การส่งสินค้า แบบ 1 เที่ยว ต่อ 1 ลูกค้า..... 8

2.2 การจัดเส้นทางเดินรถจากศูนย์กระจายสินค้าแหล่งเดียวไปยังลูกค้าต่าง ๆ..... 12

2.3 ผลลัพธ์การจัดเส้นทางเดินรถขนส่งสินค้าแบบเส้นทางเดียว (STSP)..... 13

2.4 ผลลัพธ์การจัดเส้นทางเดินรถขนส่งสินค้าแบบหลายเส้นทาง (MTSP)..... 13

2.5 ผลลัพธ์การจัดเส้นทางเดินรถขนส่งสินค้าแบบคลาสสิค (CVRP)..... 14

2.6 แสดงวิธีการรวมจุดส่งสินค้าโดยวิธี saving: Clark and Wright saving 17

2.7 การแก้ปัญหาโดยวิธี Sweep Approach 18

2.8 เทคนิคการปรับปรุงเส้นทางด้วยวิธี 2 – OPT..... 19

2.9 เทคนิคการปรับปรุงเส้นทางด้วยวิธี 3 – OPT..... 19

2.10 ระบบสารสนเทศทางภูมิศาสตร์..... 23

2.11 องค์ประกอบในการทางานของระบบ GIS 25

3.1 โปรแกรม Google Map 36

3.2 โปรแกรมที่ทำงานบน Microsoft Excel ร่วมกับ VBA (Visual Basic for Application 36

4.1 แผนภูมิการจัดเส้นทางการเดินรถด้วยวิธี 3-OPT..... 39

4.2 การเปรียบเทียบระยะทางระหว่างพนักงานจัดรถและวิธี 3-OPT..... 43

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การทำธุรกิจในปัจจุบันนี้มีการแข่งขันสูงถือเป็นเรื่องท้าทายสำหรับเจ้าของธุรกิจเป็นอย่างมาก จึงจำเป็นจะต้องหาวิธีต่าง ๆ เพื่อทำให้ธุรกิจของตนสามารถอยู่ได้ เป็นที่แน่นอนว่าธุรกิจต่าง ๆ ได้เปรียบทางการแข่งขันได้นั้น จะต้องสามารถบริการลูกค้าได้รวดเร็วกว่าคู่แข่ง ใช้ทรัพยากรที่มีอยู่ได้อย่างมีประสิทธิภาพโดยอาศัยอยู่ภายใต้ต้นทุนที่เหมาะสมหรือต่ำกว่าคู่แข่ง ทำให้ธุรกิจต่าง ๆ กันมาให้ความสนใจเกี่ยวกับการจัดการระบบให้ดีโดยการวางแผนการจัดการโลจิสติกส์ให้ดี ดังนั้นในปัจจุบัน โลจิสติกส์จึงมีบทบาทอย่างมากในกระบวนการทำงานในธุรกิจต่างๆ ไม่ว่าจะเป็นในด้านการบริหารคลังสินค้า การวางแผนระบบการผลิต และด้านการขนส่งซึ่งในปัจจุบันจะพบว่าธุรกิจหลากหลายประเภทให้บริการเสริมโดยการขนส่งสินค้าเพื่อให้ลูกค้าเลือกที่จะมาใช้บริการมากยิ่งขึ้น ดังจะเห็นจากสถิติจำนวนใบอนุญาตประกอบการขนส่งและจำนวนผู้ประกอบการขนส่งทั่วประเทศ ข้อมูล ณ วันที่ 30 มิถุนายน 2559 กรมการขนส่งทางบก มีผู้ประกอบการขนส่ง มากถึง 422, 819 ราย (ที่มา: กลุ่มสถิติการขนส่ง กองแผนงาน กรมการขนส่งทางบกปี 2559)

บริษัทกรณีศึกษาที่ผู้วิจัยได้เก็บข้อมูล เป็นบริษัทผู้ให้บริการขนส่งทางด้านโลจิสติกส์รับขนส่งสินค้าไปยังห้างสรรพสินค้าต่างๆ โดยถ้าบริษัทมีแผนการส่งสินค้าให้เกิดความคุ้มค่าในแต่ละเที่ยว และสามารถลดระยะทางในการวิ่งของรถขนสินค้า ก็จะส่งผลให้ต้นทุนการจัดส่งสินค้าลดลง ทำให้บริษัทมีโอกาสในการแข่งขันทางการค้ามากยิ่งขึ้น สำหรับปัญหาของกรณีศึกษานี้ พบว่าขั้นตอนในการส่งสินค้าให้ลูกค้าแต่ละรายนั้นยังไม่ได้นำหลักการวิชาการมาใช้ในการจัดสรรเส้นทางการเดินรถให้คุ้มค่า อาศัยเพียงความคุ้นเคยของพนักงานที่จัดเส้นทางการเดินรถเพียงอย่างเดียว โดยไม่ได้คำนึงถึงระยะทางและจำนวนสินค้าในแต่ละเที่ยว ใช้หลักการสถานที่ใกล้ไปส่งด้วยกัน โดยไม่พิจารณาหลักเกณฑ์ใดๆทั้งสิ้น อาศัยเพียงความชำนาญของพนักงานขับรถเท่านั้น ซึ่งมีความเสี่ยงต่อการใช้ทรัพยากรในแต่ละเที่ยวบรรทุกอย่างไม่คุ้มค่า

ดังนั้นจึงเป็นที่มาของงานวิจัยในการประยุกต์ใช้วิธีการเมต้าฮิวริสติกส์ ด้วยวิธี 3 OPT ซึ่งวิธีนี้เป็นวิธีหนึ่งของวิธีการเมต้าฮิวริสติกส์ สำหรับแก้ปัญหาการจัดเส้นทางการเดินรถของกรณีศึกษา ที่จะส่งสินค้าให้ลูกค้าแต่ละราย โดยมุ่งเน้นที่จะลดระยะทางการขนส่งให้มีระยะทางโดยรวมสั้นที่สุด และนำผลที่ได้มาเปรียบเทียบกับสภาพปัจจุบันเพื่อหาคำตอบ ตลอดจนสามารถใช้ทรัพยากรที่มีอยู่ได้อย่างมีประสิทธิภาพ

1.2 คำถามการวิจัย

1.2.1 อะไรเป็นเทคนิคในการจัดเส้นทางการขนส่งด้วยวิธีการเมต้าฮิวริสติกส์

1.2.2 เทคนิคในการจัดเส้นทางการขนส่งด้วยวิธีการเมต้าฮิวริสติกส์ช่วยลดระยะทางในการขนส่งให้สั้นที่สุดได้อย่างไร

1.3 วัตถุประสงค์ของการวิจัย

1.3.1 เพื่อศึกษาเทคนิคในการจัดเส้นทางการขนส่งด้วยวิธีการเมต้าฮิวริสติกส์มาใช้ในการจัดเส้นทางการขนส่งสินค้าให้กับบริษัทกรณีศึกษา

1.3.2 เพื่อศึกษาวิธีการนำเทคนิคในการจัดเส้นทางการขนส่งด้วยวิธีการเมต้าฮิวริสติกส์มาช่วยในการลดระยะทางในการขนส่งให้ได้ระยะทางที่สั้นที่สุด

1.4 ประโยชน์ที่คาดว่าจะได้รับจากวิจัย

1.4.1 วิธีการเมต้าฮิวริสติกส์สามารถลดต้นทุนการขนส่งให้กับบริษัทกรณีศึกษาได้

1.4.2 วิธีการเมต้าฮิวริสติกส์ช่วยในการจัดเส้นทางการขนส่งให้กับบริษัทกรณีศึกษาโดยให้ได้ระยะทางสั้นที่สุด

1.4.3 เป็นประโยชน์แก่บริษัทอื่น ๆ เพื่อใช้ในการลดต้นทุนการขนส่งและจัดเส้นทางการขนส่งให้ได้ระยะทางสั้นที่สุด

1.5 ขอบเขตการวิจัย

ขอบเขตด้านเนื้อหา

1.5.1 การศึกษานี้จะพิจารณาการขนส่งสินค้าของบริษัทกรณีศึกษาเฉพาะในเขตกรุงเทพและปริมณฑลเท่านั้น

1.5.2 รถที่ใช้ในการขนส่งประเภทหกล้อ สามารถบรรทุกสินค้าได้ไม่เกิน 7,500 กิโลกรัม

1.5.3 การศึกษานี้จะศึกษาวิธีการเมต้าฮิวริสติกส์ด้วยวิธีการ 3 OPT ในการลดระยะทางการขนส่งเท่านั้น

ขอบเขตด้านพื้นที่

พื้นที่ของบริษัทกรณีศึกษานี้ตั้งอยู่ บริเวณถนน บรมราชชนนี ตำบลบางเตย อำเภอสามพราน จังหวัดนครปฐม โดยผู้วิจัยจะเก็บข้อมูลโดยลงพื้นที่ไปยังกรณีศึกษา และเก็บข้อมูลการขนส่งไม่จำกัดระยะทางในแต่ละวันเป็นระยะเวลา 7 วัน

ขอบเขตด้านระยะเวลา

การศึกษานี้จะเก็บข้อมูลเฉพาะบริษัทกรณีศึกษาเพียงบริษัทเดียวเท่านั้น เนื่องจากบริษัทกรณีศึกษานี้ เป็นผู้ให้บริการขนส่งและกระจายสินค้าให้กับ ผู้ประกอบการอุตสาหกรรม ห้าง ร้าน ในเขตกรุงเทพมหานคร ปริมณฑล และต่างจังหวัด ปัจจุบันมีลูกค้าใช้บริการ ไม่น้อยกว่า 2,000 ราย และเป็นบริษัทที่ให้บริการด้านโลจิสติกส์แบบครบวงจร โดยในการเก็บข้อมูลเพื่อทำการวิจัยนี้ครั้งนี้ ในช่วงเดือน ตุลาคม 2560

1.6 นิยามศัพท์เฉพาะ

1.6.1 วิธีการเมต้าฮิวริสติกส์ หมายถึง วิธีในการค้นหาคำตอบที่ดีภายในพื้นที่ของคำตอบที่เป็นไปได้ มีจุดประสงค์เพื่อหาคำตอบที่ดีที่สุดหรือคำตอบที่ใกล้เคียงคำตอบที่ดีที่สุดภายในระยะเวลาอันสั้น แม้ว่านำไปประยุกต์ใช้ในปัญหาที่หลากหลาย

1.6.2 วิธีการสลับสามตำแหน่ง (3- OPT) หมายถึง เทคนิคการปรับปรุงเส้นทางภายในเส้นทางเดียวกันที่มีความคล้ายคลึงกับเทคนิคการปรับปรุงเส้นทางด้วยวิธี 2-Opt แต่การสลับลำดับจุดส่งนั้นจะถูกเลือกสลับครั้งละ 3 จุด

1.6.3 การขนส่ง หมายถึง กระบวนการเคลื่อนย้ายสินค้าและบริการจากผู้ผลิตไปสู่ผู้บริโภค หรือเป็นการเคลื่อนย้ายวัตถุดิบไปยังโรงงานเพื่อทำการผลิตสินค้าหรือบริการ โดยใช้เครื่องมือและอุปกรณ์ในการขนส่งเป็นพาหนะพาไปตามความต้องการและเกิดอรรถประโยชน์ตามที่ผู้ทำการขนส่งต้องการ

1.6.4 การจัดเส้นทางการขนส่ง หมายถึง การกำหนดเส้นทางของยานพาหนะแต่ละคันเพื่อไปให้บริการลูกค้าที่กำหนดโดยเสียค่าใช้จ่ายในการขนส่งน้อยที่สุด

1.6.5 การขนส่งสินค้าจากจุดเดียว หมายถึง การขนส่งสินค้าจากคลังสินค้าเพียงแห่งเดียวไปยังลูกค้ารายต่างๆ

บทที่ 2
แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

การวิจัยครั้งนี้มีวัตถุประสงค์ในการวิจัยเพื่อประยุกต์ใช้วิธีการเมต้าฮิวริสติกส์ในการจัดเส้นทางการขนส่งสินค้าและนำมาใช้เป็นเครื่องมือในการลดต้นทุนการขนส่งสินค้าและเพิ่มประสิทธิภาพในการขนส่งให้กับบริษัทกรณีศึกษา ผู้วิจัยนำเสนอแนวคิดทฤษฎีและงานวิจัยที่เกี่ยวข้องโดยแบ่งออกตามประเด็นดังต่อไปนี้

2.1 ความหมายของการขนส่ง

2.2 รูปแบบของการขนส่งสินค้า

2.3 ลักษณะของการขนส่งสินค้าด้วยรถบรรทุก

2.4 การจัดเส้นทางการเดินรถ (Vehicle Routing)

2.5 ปัญหาการจัดเส้นทางการเดินรถ

2.6 เทคนิคการหาคำตอบ

2.7 ระบบสานสนเทศทางภูมิศาสตร์

2.8 Visual Basic วิชวลเบสิก

2.9 กรณีศึกษา บริษัทผู้ให้บริการขนส่งทางด้านโลจิสติกส์

2.10 งานวิจัยที่เกี่ยวข้อง

โดยมีรายละเอียดดังนี้

2.1 ความหมายของการขนส่ง

แลมเบิร์ต, สต็อค และเอลแรม (2546: 137) อธิบายว่า การขนส่ง หมายถึง การเคลื่อนย้ายสินค้าจากสถานที่ผลิตไปยังสถานที่บริโภคสินค้านั้น ซึ่งการเคลื่อนย้ายสินค้าระหว่างสถานที่ ดังกล่าวก่อให้เกิดมูลค่าเพิ่มแก่สินค้า ซึ่งมูลค่าเพิ่มนี้เรียกว่า อรรถประโยชน์ด้านสถานที่ (Place Utility) นอกจากนี้การขนส่งยังก่อให้เกิดอรรถประโยชน์ด้านเวลา (Time Utility) ซึ่งเกิดขึ้นจากการ เก็บรักษาสินค้าไว้จนกระทั่งเกิดความต้องการในการบริโภคสินค้านั้น

วิโรจน์ พุทธวิถี (2547: 170) อธิบายว่า การขนส่งเป็นการเคลื่อนย้ายสินค้าและบริการจากแหล่งผู้ผลิตหรือผู้จัดเก็บ ไปยังลูกค้าในระดับต่าง ๆ การขนส่งจะเป็นตัวขับเคลื่อนสินค้าหรือ บริการเข้าไปยังกลุ่มเป้าหมายทางการตลาด ทำให้สินค้าและบริการถูกส่งมอบไปในที่ที่ต้องการ บริโภค และสินค้าหรือบริการนั้นจะเป็นที่รู้จักมากขึ้น

จากข้อมูลข้างต้นสรุปได้ว่า การขนส่ง คือ การเคลื่อนย้ายบุคคลหรือสิ่งของจากสถานที่หนึ่งไปยังอีกสถานที่หนึ่ง ซึ่งก่อให้เกิดอรรถประโยชน์ด้านสถานที่และเวลา และมีรูปแบบในการขนส่งที่หลากหลาย ดังจะกล่าวในหัวข้อถัดไป

2.2 รูปแบบของการขนส่งสินค้า

ในด้านกายภาพ (Physical view) การขนส่งสินค้า สามารถแบ่งออกได้เป็น 5 รูปแบบ (วิโรจน์ พุทธวิถี, 2547: 173-176) ได้แก่

2.2.1 การขนส่งสินค้าทางถนน (Road transportation) คือการขนส่งที่ใช้รถยนต์ในการบรรทุกสินค้า และขนส่งระหว่างตำแหน่งต่าง ๆ ที่มีแผ่นดินเชื่อมกัน การขนส่งเกือบทุกกรณีจะต้องอาศัยการขนส่งทางถนน เพราะเป็นการขนส่งที่สามารถเข้าถึงต้นทางและปลายทางได้อย่างสะดวกรวดเร็ว

2.2.2 การขนส่งสินค้าทางราง (Rail transportation) เป็นการขนส่งผ่านระบบรางที่มีอุปกรณ์หลัก คือ ขบวนรถไฟ ระบบนี้มีข้อจำกัดในด้านสถานที่ตั้งของตำแหน่งสถานี ที่มีเฉพาะเท่าที่ภาครัฐของประเทศนั้น ๆ สร้างไว้เท่านั้น ซึ่งไม่ครอบคลุมทุกที่ แต่เป็นรูปแบบการขนส่งที่มีต้นทุนต่ำกว่าการขนส่งสินค้าทางถนน โดยเหมาะสำหรับการขนส่งสินค้าปริมาณมากและในระยะทางไกล

2.2.3 การขนส่งสินค้าทางน้ำ (Water transportation) เป็นรูปแบบการขนส่งที่ประหยัดที่สุด โดยจะใช้เรือสินค้าที่มีขนาดใหญ่ในการขนส่งสินค้าทำให้ขนส่งได้ครั้งละมาก ๆ แต่มีความเร็วต่ำจึงเป็นระบบขนส่งหลักของการขนส่งระหว่างประเทศ

2.2.4 การขนส่งสินค้าทางอากาศ (Air transportation) เป็นรูปแบบการขนส่งที่ใช้เครื่องบินในการขนส่งสินค้า จึงมีความรวดเร็ว และค่าขนส่งที่สูง และสินค้ามีความปลอดภัยกว่าการขนส่งในรูปแบบอื่น

2.2.5 การขนส่งทางท่อ (Pipeline transportation) เป็นการขนส่งซึ่งใช้สาหรับการขนส่งของเหลว โดยมีการกำหนดที่ตั้งของสถานที่ส่งและรับแน่นอนและใช้ในการขนส่งที่สร้างขึ้น เฉพาะเท่านั้น เช่น ปิโตรเลียม น้ำประปา ก๊าซธรรมชาติ

จากรูปแบบการขนส่งสินค้าที่ได้กล่าวมาแล้วนั้น แต่ละรูปแบบการขนส่งมีความเหมาะสมกับประเภทสินค้าและลักษณะการขนส่งที่แตกต่างกันไป ขึ้นอยู่กับคุณลักษณะของการขนส่งสินค้า ในแต่ละรูปแบบ ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 เปรียบเทียบคุณลักษณะของรูปแบบการขนส่งต่าง ๆ

คุณลักษณะ	ทางถนน	ทางราง	ทางน้ำ	ทางอากาศ	ทางท่อ
ความเร็วการขนส่ง	2	3	4	1	5
ความมีพร้อมบริการ	1	2	4	3	5
ความเร็วของระเบียบขั้นตอน	2	3	4	5	1
ปริมาณที่ขนส่งได	3	2	1	4	5
ความยืดหยุ่นตามความต้องการ	2	4	5	3	1
รวม	**10**	**14**	**18**	**16**	**17**

หมายเหตุ 1 = ดีมาก, 2 = ดี, 3 = ปานกลาง, 4 = พอใช้ และ 5 = ไม่ดี

(วิโรจน์ พุทธวิถี, 2547: 178)

จากตารางที่ 2.1 สรุปได้ว่า การขนส่งสินค้าทางถนนถือเป็นการขนส่งที่สะดวกที่สุด เนื่องจากสามารถเข้าถึงได้ทุกจุด ถึงแม้ว่าจะมีต้นทุนการขนส่งสูงกว่าการขนส่งในรูปแบบอื่น ๆ

2.3 ลักษณะของการขนส่งสินค้าด้วยรถบรรทุก

ลักษณะของการขนส่งสินค้าด้วยรถบรรทุกจำแนกตามลักษณะของการบรรทุกสินค้า สามารถจำแนกได้ 2 ลักษณะ คือ

การขนส่งสินค้าแบบเต็มคันรถ (Full Truckload or FTL Freight) ชารอน และ ซาน (Barker Sharon and San, 1981) ได้นิยามการขนส่งแบบเต็มคันรถว่า เป็นลักษณะหนึ่งของการขนส่งแบบว่าจ้าง ซึ่งต้องขนส่งสินค้าจากสถานที่หนึ่งไปยังอีกสถานที่หนึ่งที่มีจำนวนสินค้าของลูกค้าเท่ากับจำนวนสินค้าที่รถบรรทุกสามารถบรรทุกได้

คมกฤช วลีวงศ์ (2547) ได้นิยามการขนส่งแบบเต็มคันรถว่า หมายถึง การขนส่งสินค้าที่ปริมาณสินค้าที่บรรทุกในรถแต่ละเที่ยวเต็มขีดจากัดของความจุที่กำหนดไว้ และมักมีการขนส่ง สินค้าทั้งหมดไปยังจุดหมายเพียงจุดเดียว

การขนส่งสินค้าแบบไม่เต็มคันรถ (Less-Than-Truckload or LTL Freight) โพเวลล์ และเซฟฟี (Powell and Sheffi, 1983) ได้นิยามการขนส่งแบบไม่เต็มคันรถว่าเป็นการขนส่งสินค้าที่จำนวนสินค้าของลูกค้าในที่ต่าง ๆ มีน้อยกว่าจำนวนสินค้าที่รถบรรทุกสามารถบรรทุกได้ ดังนั้นจึงต้องมีการรวมสินค้าจากลูกค้าหลาย ๆ แหล่ง (Consolidation) เพื่อให้สินค้าเต็มคันรถทั้งนี้เพื่อลดค่าใช้จ่ายในการขนส่งสินค้า

คมกฤช วลีวงศ์ (2547) ได้นิยามการขนส่งแบบไม่เต็มคันรถว่า หมายถึง การขนส่งสินค้าที่มีการขนส่งในแต่ละเที่ยวรถไม่ถึงขีดจำกัดของรถที่กำหนดไว้ ทำให้อัตราค่าขนส่งสูงกว่ารูปแบบการขนส่งแบบเต็มคันรถ

จากข้อมูลข้างต้นสรุปได้ว่า การขนส่งสินค้าแบบเต็มคันรถเป็นการขนส่งสินค้าที่ปริมาณสินค้าของลูกค้าเท่ากับขีดจำกัดของปริมาณสินค้าที่รถบรรทุกบรรทุกสินค้าได้ และการขนส่งสินค้าแบบไม่เต็มคันรถ คือ เป็นการขนส่งสินค้าที่ปริมาณสินค้าของลูกค้าน้อยกว่าขีดจำกัดของปริมาณสินค้าที่รถบรรทุกบรรทุกสินค้าได้

2.4 การจัดเส้นทางการเดินรถ (Vehicle Routing)

เป็นปัญหาประจำของบริษัทขนส่ง ในแต่ละวันจะมีลูกค้าจำนวนหนึ่งมากบ้างน้อยบ้างต้องการให้นาสินค้าไปส่งให้ ปัญหาที่น่าท้าทายคือผู้จัดการจะต้องใช้รถขนส่งกี่คันและควรจะจัดลำดับการส่งสินค้าอย่างไร รถคันไหนควรไปส่งสินค้าให้แก่ลูกค้ารายใดบ้าง และจะจัดลำดับการส่งสินค้าของลูกค้าแต่ละรายอย่างไร ในทางคณิตศาสตร์แล้วถือว่าปัญหาการจัดเส้นทางการเดินรถ (Vehicle Routing Problem) เป็นปัญหาที่ยากมากๆ ในการที่จะวิเคราะห์หาแผนการเดินรถที่ดีที่สุดในบรรดาแผนที่เป็นไปได้จำนวนมากโดยเฉพาะอย่างยิ่งหากคำสั่งซื้อจากลูกค้าและรถส่งสินค้ามีจำนวนมาก ซึ่งแทบจะไม่มีโอกาสเลยที่จะจัดเส้นทางการเดินรถให้ประหยัดที่สุด อย่างดีที่สุดคงทำได้เพียงการวิเคราะห์ให้ได้แผนที่ค่อนข้างดีมาใช้ปฏิบัติเท่านั้นในกรณีที่มีลูกค้าจำนวนมาก การหาเส้นทางที่เป็นไปได้ทั้งหมดแล้วเลือกเอาเส้นทางขนส่งที่มีต้นทุนต่ำที่สุด เป็นสิ่งที่แทบจะเป็นไปไม่ได้ในทางปฏิบัติ เพราะจะต้องใช้เวลาในการคำนวณวิเคราะห์ยาวนานมากจนไม่ทันการณ์ จึงจำเป็นต้องมีวิธีการอื่นๆ มาช่วยให้ได้คำตอบที่เร็ว ซึ่งมีวิธีการหนึ่งที่เป็นที่นิยมใช้กันอย่างแพร่หลายในการจัดเส้น ทางการเดินรถ ได้แก่ การจัดเส้นทางโดยค่าประมาณ (Approximation Methods) ซึ่งคำตอบของเส้นทางขนส่งที่ได้รับจากการใช้วิธีการนี้อาจจะไม่ดีที่สุดแต่จะได้คำตอบที่ดีพอใช้ ภายในระยะเวลาที่ไม่นานมากนำไปใช้งานได้ทันเวลา วิธีการจัดเส้นทางโดยค่าประมาณนี้ยังมีหลากหลายวิธี คำตอบที่ได้รับจากแต่ละวิธีอาจจะมีคุณภาพแตกต่างกัน ขึ้นอยู่กับลักษณะรายละเอียดของปัญหาแต่ละกรณี ซึ่งวิธีที่ได้รับความนิยมและเข้าใจง่ายวิธีการหนึ่งคือวิธีเซฟวิ่งอัลกอริทึม (Saving Algorithm) ซึ่งเสนอโดย Clarke and Wright นักวิจัยในประเทศอังกฤษ ใน ค.ศ. 1964 ซึ่งได้พิจารณาการจัดเส้นทางยานพาหนะที่มีความต้องการของลูกค้าหลายราย และยานพาหนะมีความจุหลายขนาดส่งสินค้าออกจากคลังพัสดุแห่งเดียว งานวิจัยนี้ได้พัฒนาขั้นตอนให้สามารถเลือกเส้นทางยานพาหนะที่เหมาะสมที่สุด และผลที่ได้จากการแก้ปัญหานี้คือ ทำให้ทราบจำนวนยานพาหนะที่จะใช้ในการขนส่ง และปริมาตรสินค้าที่ขนส่งโดยยานพาหนะแต่ละคัน โดยมีวิธีในการดำเนินงานดังนี้

1. เลือกจุดเริ่มต้นจากคลังสินค้าขึ้นมาหนึ่งปุ่มให้เป็นปุ่มที่หนึ่ง
2. คำนวณค่าของระยะเวลา, ระยะทางหรือค่าใช้จ่ายในการขนส่งที่ประหยัด (Saving Cost), Sij = CiD + CDj – Cij เมื่อ i , j คือลูกค้า และ D คือคลังสินค้า
3. เรียงลำดับค่า Sij จากมากไปหาน้อย
4. สร้างเส้นทางของยานพาหนะโดยเชื่อมปุ่ม i และ j ที่มีค่า Sij มากที่สุด
5. ทำซ้ำจนกว่าจะจัดเส้นทางได้ครบ โดยมีเงื่อนไขของข้อจากัดในการเดินทางแต่ละยานพาหนะจะต้องมีสินค้าไม่เกินความจุของยานพาหนะ และต้องใช้เวลาในการเดินทางไม่เกินระยะเวลาที่กำหนด

วิธีเซฟวิ่งอัลกอริทึมเป็นทฤษฎีที่เป็นที่ยอมรับในการจัดการปัญหาการขนส่งยานพาหนะ ใจความของทฤษฎีไม่ซับซ้อน คือ พิจารณาการส่งจากคลังสินค้า D

ภาพที่ 2.1 การส่งสินค้า แบบ 1 เที่ยว ต่อ 1 ลูกค้า

จากรูปที่ 1 ถ้าใช้รถ 1 คัน วิ่งส่งสินค้าให้ลูกค้า 2 ราย (i และ j) ในเที่ยวเดียวกันระยะทางทั้งหมดจะลดลงเท่ากับ S(i, j) = 2d(D, i) + 2d(D, j) – [d(D, i) + d(i, j) + d(D, j)] = d(D, i) + d(D, j) – d(i, j) ค่า Saving S(i, j) ที่ได้คือระยะทางที่สามารถลดได้ หากระยะทางระหว่างลูกค้าใด ทำให้เกิดค่า Saving สูงก็หมายความว่า สามารถลดระยะทางได้มาก

โปรแกรมเชิงเส้นตรง (Linear Programming) เป็นเทคนิคที่รู้จักกันแพร่หลายและเป็นส่วนหนึ่งของการวิจัยดำเนินงาน(Operations Research) ในหลาย ๆ หน่วยงานได้ประยุกต์ใช้วิธีการทางโปรแกรมเชิงเส้น ในการแก้ปัญหาทางการจัดสรรปัจจัยหรือทรัพยากร (allocating resource) โดยที่ปัจจัยหรือทรัพยากรมีความหมายรวมถึงวัตถุดิบ กำลังคน เวลา สถานที่ เงิน หรือความรู้ความสามารถต่างๆ โดยมีลักษณะความสัมพันธ์ของตัวแปรต่างๆเป็นแบบเชิงเส้น มีจุดหมายเพื่อแก้ปัญหาและตัดสินใจให้เกิดผลตามแนวทางการดำเนินงานที่ดีที่สุด (OPTimal) เช่น กำไรสูงสุด ค่าใช้จ่ายน้อยที่สุด หรือแนวทางการดำเนินงานอื่นๆที่ให้ผลประโยชน์มากที่สุดต่อระบบนั้น ๆ โดยพิจารณาเงื่อนไขหรือข้อจากัดที่กำหนด เช่น สภาวะตลาด การขาดแคลนวัตถุดิบ กำลังคนเงินทุน

สถานที่ ความรู้ข้อกำหนดของกฎหมายและระเบียบต่างๆของสังคม นโยบายของฝ่ายบริหาร ขอบข่ายของธุรกิจที่ดำเนินอยู่และอื่นๆ เทคนิคทางการโปรแกรม เชิงเส้นนี้พัฒนามาจากผลความก้าวหน้าทางวิทยาศาสตร์ ซึ่งมีแนวคิดริเริ่มมาจากนักคณิตศาสตร์ วิทยาศาสตร์หลาย ๆ ท่านซึ่งได้นาไปใช้ในทฤษฎีเกม รวมทั้งถูกพัฒนานาไปใช้ในทางการขนส่ง แก้ปัญหาทางโภชนาการ และแก้ปัญหาทางการวางแผนโครงงานในกองทัพ ปัจจุบันเป็นที่ยอมรับกันในหลายๆวงการในการนาเทคนิคทางการโปรแกรมเชิงเส้นไปใช้ประโยชน์ในหลาย ๆ ด้าน เช่น ทางการเกษตร ทางเศรษฐศาสตร์ และการจัดการเกี่ยวกับการผลิตทางอุตสาหกรรม โปรแกรมเชิงเส้นประกอบไปด้วย 2 ส่วนดังนี้

1. มีสมการกำหนดเป้าหมาย (objective function) คือสมการแสดงความสัมพันธ์ของต้นทุน กำไร เพื่อให้กำหนดเป้าหมายสูงสุดหรือต่ำสุด
2. มีสมการแสดงขอบข่าย (constraints) ซึ่งแสดงข้อจากัดต่างๆของปัจจัยหรือทรัพยากรในรูปสมการหรืออสมการโดยที่สมการต่างๆทั้งหมดเป็นสมการเชิงเส้น

2.5 ปัญหาการจัดเส้นทางการเดินรถ

การจัดเส้นทางในการเดินรถที่ดีและมีประสิทธิภาพนั้น จะส่งผลให้สามารถลดระยะทางในการขนส่งได้ และในบางครั้งยังสามารถลดจำนวนพยานพาหนะที่ใช้ในการขนส่งได้อีกด้วย เมื่อระยะทางและจำนวนยานพาหนะที่ใช้ในการขนส่งลดลง ส่งผลให้มลพิษที่เกิดขึ้นจากการขนส่ง ลดลง และยังทำ ให้ต้นทุนในการขนส่งของบริษัทลดลงอีกด้วย ซึ่งรูปแบบในการจัดเส้นทางการขนส่งสามารถแบ่งได้เป็น 13 รูปแบบ (ศิริวรรณ โพธิ์ทอง: ออนไลน์ 2553) โดยมีรายละเอียด ดังนี้

1. การหาเส้นทางที่สั้นที่สุด (Shortest Path) เป็นการจัดเส้นทางการขนส่งโดยให้มีการ ขนส่งสินค้าจากบริษัทไปยังลูกค้าทุกๆ รายให้มีระยะทางรวมในการขนส่งที่น้อยที่สุด โดยหา เส้นทางที่มีระยะทางสั้นสุดระหว่างจุดเริ่มต้นกับจุดหมาย (Thomas, Charles and Ronald, 1990)

2. ปัญหาการเดินทางของเซลส์แมน (Traveling Salesman Problem: STP) เป็นการจัดเส้นทางการขนส่งโดยให้มีการขนส่งสินค้าจากบริษัทไปยังลูกค้าทุกๆ รายให้มีระยะทางรวมในการขนส่งที่น้อยที่สุด โดยรวมระยะทางการเดินทางของพาหนะที่ใช้ในการขนส่งสินค้ากลับมายังบริษัทด้วย

3. ปัญหาการจัดเส้นทางสาหรับยานพาหนะ (Vehicle Routing Problem: VRP) เป็นการจัดเส้นทางในการขนส่งสินค้าโดยหาจำนวนพาหนะที่ใช้ในการขนส่งสินค่า ที่เหมาะสมกับปริมาณสินค้า พร้อมทั้งหาเส้นทางเส้นทางในการขนส่งสินค้าที่ เหมาะสม ภายใต้ข้อกำหนดในเรื่องของ ค่าใช้จ่ายที่น้อยที่สุด

4. ปัญหาการขนส่ง (Transportation Problem) เป็นการจัดเส้นทางในการขนส่งสินค้า จากคลังสินค้าหลายแห่งไปยังลูกค้าหลายราย โดยการปริมาณในการขนส่งสินค้าต้องเท่ากับปริมาณ ความต้องการของลูกค้าที่สั่ง ไว้และเสียค่าใช้จ่ายในการขนส่งน้อยที่สุด

5. การไหลของค่าใช้จ่ายอย่างต่ำสุด (Minimum Cost Flow) เป็นการจัดเส้นทางในการ ขนส่งสินค้าจากคลังสินค้าไปยังลูกค้าแต่ละราย ตามปริมาณและจำนวนที่ลูกค้าต้องการและมี ค่าใช้จ่ายในการขนส่งต่ำ ที่สุด

6. ต้นไมแบบทอดขามเล็กสุด (Minimum Spanning Tree) เป็นการจัดเส้นทางการขนส่งสินค้าโดยพิจารณาถึงความเชื่อมโยงของเส้นทางที่ ใช้ในการขนส่ง โดยระยะทางรวมในการขนส่งน้อยที่สุด โดยวิธีนี้ถูกนำไปประยุกต์ใช้ในงานวิจัยหลาย ๆ ด้าน

7. การเลือกทำเลที่ตั้ง (Facility Location) เป็นการหาตำแหน่งที่ตั้งที่เหมาะสมของจุดกระจายสินค้า เพื่อให้การขนส่งสินค้าจากบริษัทไปยังลูกค้ามีระยะทางและค่าใช้จ่ายในการขนส่งน้อยที่สุด

8. การบรรจุสิ่งของลงกล่อง (Bin Packing) เป็นวิธีการจัดจำนวนพาหนะที่น้อยที่สุดที่ใช้ในการขนส่งสินค้าให้เพียงพอกับปริมาณสินค้าที่ต้องส่งไปให้ลูกค้า

9. การจัดตารางเครื่องจักรโหลดบรรจุภัณฑ์ (Parallel Machine Scheduling) เป็นการจัดเส้นทางในการขนส่งสินค้าโดยคำนึงถึงลา ดับขั้นตอนในการดา เนิน การโหลดสินค้าไปยังพาหนะที่ ใช้ในการขนส่งเพื่อให้พาหนะที่เสร็จเป็นลา ดับสุดท้ายใช้เวลาน้อยที่สุด

10. การจัดตารางงานของพนักงาน (Crew Scheduling) เป็นวิธีการจัดตารางทำงานของพนักงานที่ดำเนินงานเกี่ยวกับการขนส่ง เช่น พนักงานขับรถ พนักงานโหลดสินค้า

11. การวางแผนขนสินค้าบนเครื่องบิน (Aircraft Load Planning) เป็นวิธีการจัดแผนการวางสัมภาระในท้องเครื่องบินให้สมดุล เพื่อความปลอดภัยและประหยัดน้ำมัน

12. การจัดแผนการเคลื่อนย้ายตู้คอนเทนเนอร์ในท่าเรือ (Container Handling at Ports) การวางแผนการเคลื่อนย้ายตู้คอนเทนเนอร์ไปมาระหว่างเรือกับท่าเรือ และ ท่าเรือกับรถบรรทุก

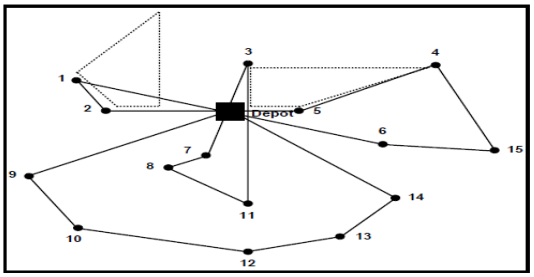
13. วิธีศึกษาสำนึก (Heuristics) เป็นวิธีการแก้ปัญหาที่ใช้ในการค้นหาคำตอบที่ใกล้เคียงคำตอบที่เหมาะสมที่สุด ซึ่งรูปแบบการแก้ปัญหาและการค้นหาคำตอบ จะเป็นวิธีการคิดอย่างมีเหตุผลซึ่งอาศัยการกำหนดกฎเกณฑ์บางประการขึ้นมา เพื่อหาคำตอบที่ดีและเหมาะสมในระดับหนึ่ง ถึงแม้อาจไม่ใช่คำตอบที่ดีที่สุดแต่จะได้คำตอบที่รวดเร็วโดยคำตอบที่ได้นั้นจะต้องเป็นคำตอบที่ดีเพียงพอและยอมรับได้ และวิธีการแก้ปัญหาจะพัฒนาขึ้นตามระดับความยากง่ายของปัญหาซึ่งจะนำความคิดสามัญสำนึกของมนุษย์ผนวกเข้ากับแบบจำลองทางคณิตศาสตร์เพื่อใช้ในการแก้ปัญหา โดยใช้วิธีการแบ่งพื้นที่ออกเป็นกลุ่ม (Cluster First – Route Second) แล้วทำการจัดเส้นทางเดินรถโดยใช้วิธีเลือกจุดที่ใกล้ที่สุด (Nearest Neighbor Approach)

ปัญหาการจัดเส้นทางเดินรถ (Vehicle Routing Problem : VRP) ในอดีตระยะสามสิบถึงสี่สิบปีที่ผ่านมา ปัญหานี้ได้รับความสนใจและมีการศึกษากันอย่างกว้างขวางทุกแง่มุมที่ หลากหลาย รวมถึงมีความซับซ้อนใกล้เคียงกับปัญหาที่พบได้ในสถานการณ์จริงมากยิ่งขึ้นการพัฒนาวิธีการ และแนวทางการแก้ไขปัญหามีหลากหลายโดยมีจุดเริ่มต้นจากบทความของ Dantzig and Ramser (1959) ซึ่งมีการตีพิมพ์ในช่วงทศวรรษ 1950 ด้วยเหตุที่ปัญหานี้ได้รับความสนใจเป็นอย่างมาก เนื่องจากปัญหาการจัดเส้นทางเดินรถนั้นเป็นปัญหาที่พบได้บ่อยในชีวิตประจำวัน และปัญหาการจัดเส้นทางเดินรถเป็นปัญหาที่มีความสนใจในเชิงทฤษฎีและก็ไม่ง่ายในการที่จะหาคำตอบ ซึ่งตัวปัญหาจะถูกพิจารณาแบ่งออกเป็น 2 กลุ่มหลัก ๆ คือกลุ่มของปัญหาที่มีลักษณะเป็นแบบ Deterministic หมายถึงปัญหาที่ต้องทราบข้อมูลที่จำเป็นเบื้องต้นก่อนค่อยเริ่มทำการจัดเส้นทาง ส่วนกลุ่มของปัญหาที่มีลักษณะเป็นแบบ Dynamic จะเป็นปัญหาที่ไม่ทราบข้อมูลเบื้องต้นก่อนทำการการจัดเส้นทาง แต่จะมีข้อมูลทยอยออกมา ในระหว่างที่ทำการจัดเส้นทาง โดยปัญหาพื้นฐานของการจัดเส้นทางเดินรถที่รู้จักและมี การศึกษาวิจัยกันมากที่สุดคือ ปัญหาการเดินทางของพนักงานขาย (The Traveling Salesman Problem : TSP) ซึ่งมีจุดเริ่มต้นในช่วงทศวรรษที่ 1920 โดยนักคณิตศาสตร์และนักเศรษฐศาสตร์ ที่มีชื่อว่า Karl Menger จากนั้นก็มีการศึกษาค้นคว้ากันเรื่อยมาทำให้ปัญหานี้ได้รับความนิยม และเป็นที่รู้จักกันอย่างแพร่หลายโดยนักคณิตศาสตร์ชื่อ Merill Flood ต่อมาในปี ค.ศ. 1954 George Dantzig Ray Fulkerson และ Selmer Johnson ได้เสนอวิธีการหาคำตอบในการจัดเส้นทาง ซึ่งสามารถจัดการกับปัญหาการจัดเส้นทางในการเดินทางซึ่งมีเมืองที่ต้องเดินผ่านทั้งหมด 49 เมือง ได้จากนั้นก็มีการศึกษาอย่างต่อเนื่องจนกระทั่งในช่วงปี ค.ศ. 2004 Applegate Bixby Chvatal Cook และ Helsgaun สามารถหาเส้นทางที่เหมาะสมในการเดินทางผ่านเมืองทั้งหมด 24,978 เมืองในประเทศสวีเดนได้

สำหรับปัญหาการจัดเส้นทางการเดินรถ (The Vehicle Routing Problem: VRP) นั้นถือได้ว่าเป็นกรณีทั่วไปของปัญหาการเดินทางของพนักงานขาย (TSP) จะมียานพาหนะที่ใช้เดินทางมากกว่าหนึ่งคันโดยตัวปัญหาจะถูกแบ่งออกเป็นประเภทย่อย ๆ อีกหลายประเภทตามลักษณะเฉพาะของตัวปัญหา ในระยะแรกนั้นจะมีการศึกษาตัวปัญหาที่มีลักษณะเป็นแบบ Deterministic โดยเริ่มจากปัญหาการจัดเส้นทางเดินรถพื้นฐานซึ่งตัวปัญหาจะเป็นการจัดเส้นทางเดินรถให้ผ่านไปยังจุดรับให้ครบทุกจุดด้วยค่าใช้จ่ายในการเดินทางต่ำที่สุด และเมื่อมีข้อจำกัดในเรื่องขนาดความจุของยานพาหนะที่ใช้ขนส่ง ตัวปัญหาจึงถูกเรียกว่า Capacitated VRP (CVRP) แต่ถ้ามีท่าจอดรถมากกว่าหนึ่งแห่งจะถูกจัดเป็นปัญหาเรียกว่า Multiple Depot VRP (MDVRP) ส่วนปัญหาที่มีการไปส่งและรับคืนสินค้าจะถูกจัดเป็นปัญหาเรียกว่า Vehicle Routing Problem with Pick-up and Delivering (VRPPD) และถ้ามีข้อจำกัดเรื่องกรอบ ระยะเวลาในการขนส่งปัญหานี้เรียกว่า VRP with

Time Windows (VRPTW) และที่ผ่านมาก็มีผลงานวิจัยนำเสนออีกมากมายทั้งที่เป็นของคนไทยและต่างประเทศ

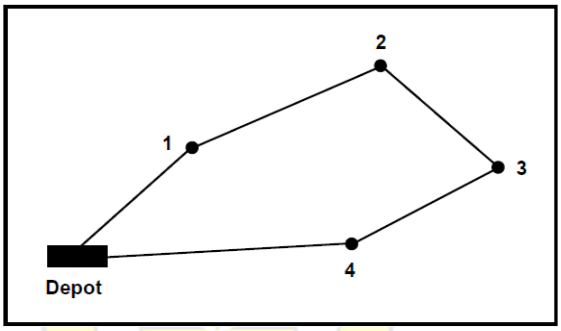
ปัญหาการเดินทางของพนักงานขาย (The Traveling Salesman Problem: TSP) เป็นปัญหาในการค้นหาเส้นทางรวมที่สั้นที่สุดที่จะใช้ในการเดินทางของพนักงานขายที่ต้องเดินทางไปยังเมืองต่าง ๆ ให้ครบทุกเมืองตามที่กำหนดโดยอย่างน้อยจะต้องเดินทางผ่านเมืองละหนึ่งครั้งซึ่งในระหว่างเมืองแต่ละเมืองจะมีเส้นทางที่เชื่อมต่อถึงกัน (Gen and Cheng, 1997) ในปี ค.ศ.1950 มีการตีพิมพ์วารสารเกี่ยวกับปัญหาการเดินทางของพนักงานขายเป็นจำนวนมากที่ได้กล่าวถึงผลงานการวิจัยที่สำคัญต่าง ๆ ในยุคนั้น และในช่วง 20 ปีที่ผ่านมานี้ได้มีวิวัฒนาการอย่างมากในการจัดการกับปัญหาการเดินทางของพนักงานขาย และได้กำหนดเป้าหมายในการวิจัยโดยใช้จำนวนเมืองที่สามารถจัดเส้นทางเดินผ่านได้ เช่น 48-City Problem, 318-City Problem, 666-City Problem และ 2392-City Problem ดังนั้นการค้นหาเส้นทางและหรือวิธีการแก้ไขปัญหาที่มีรูปแบบคล้ายกับปัญหาการจัดเส้นทางเดินรถ (The Vehicle Routing Problem: VRP) ซึ่งก็เป็นปัญหาของการจัดการเพื่อหาจำนวนเส้นทาง และจัดลำดับของการเดินรถที่มีความเหมาะสมไปยังลูกค้าต่าง ๆ ในแต่ละเส้นทางเพื่อให้สอดคล้องกับวัตถุประสงค์และเป้าหมายธุรกิจ เพราะการจัดเส้นทางเดินรถในทางปฏิบัติ แสดงดังภาพที่ 2.2 (Ballou, 1999) จะต้องคำนึงถึงปัจจัยต่างๆ ที่ส่งผลกระทบต่อการจัดเส้นทางเดินรถ (Hall and Partyka, 1997) เช่น ข้อจำกัดเส้นทางเดินรถ (Route Capacities) จะสะท้อนถึงขนาดของรถและเงื่อนไขเวลาในการขับขี่ซึ่งต้องเป็นไปตามที่กฎหมายกำหนดหรืออนุญาต กรอบเวลาเป็นการกำหนดช่วงเวลาการลงสินค้าของแต่ละลูกค้า โดยข้อกำหนดนี้อาจจะตกลงกันซึ่งเป็นไปได้ทั้งข้อกำหนดที่มีความเข้มงวดหลีกเลี่ยงไม่ได้ (Hard Time Window) และข้อกำหนดที่ไม่มีความเข้มงวดสามารถยืดหยุ่นได้ (Soft Time Window) เช่นการเสียแค่ค่าปรับหรือลงโทษ เป็นต้น



ภาพที่ 2.2 การจัดเส้นทางเดินรถจากศูนย์กระจายสินค้าแหล่งเดียวไปยังลูกค้าต่าง ๆ

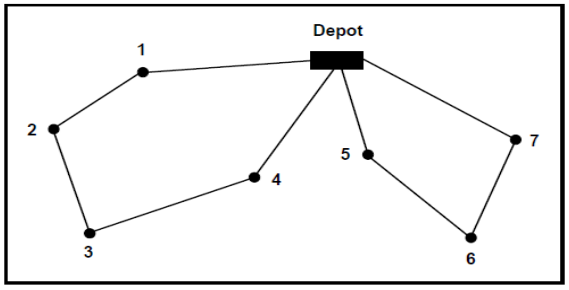
ปัญหาการจัดเส้นทางเดินรถออกจากศูนย์กระจายสินค้า (Depot) เป็นปัญหาในการจัดการเส้นทางและหาลำดับในการเดินทางไปยังลูกค้าโดยมีเป้าหมายทางธุรกิจเพื่อให้เกิดค่าใช้จ่ายน้อยที่สุด ซึ่งแนวทางในการวิเคราะห์ปัญหามีรูปแบบโดยวิธีการจำลองเรียงลำดับของปัญหาจากที่มีความซับซ้อนน้อยไปยังปัญหาที่มีความซับซ้อนมาก ซึ่งสามารถอธิบายรายละเอียดได้ดังนี้

2.5.1 การจัดเส้นทางเดินรถขนส่งสินค้าแบบเส้นทางเดียว (The Single Traveling Salesman Problem: STSP) เป็นปัญหาการจัดเส้นทางเดินรถระดับง่ายสุดเนื่องจากเป็นการจัดลำดับการขนส่งสินค้าแบบใช้เส้นทางเดินรถทางเดียวเพื่อขนส่งสินค้าให้กับลูกค้าตามจุดต่างๆโดยรถบรรทุกจะเดินทางออกจากศูนย์กระจายสินค้าเดียว ไม่มีข้อจำกัดทางด้านเวลา และขนาดความจุบรรทุกของรถ ซึ่งผลลัพธ์ของเส้นทางที่จัดได้จะเริ่มต้นเดินทางผ่านจุดของลูกค้าแต่ละรายเพียงครั้งเดียว และเดินทางวนรอบมาสิ้นสุดที่ศูนย์กระจายสินค้าเช่นเดิม ดังตัวอย่างแสดงในภาพที่ 2.3



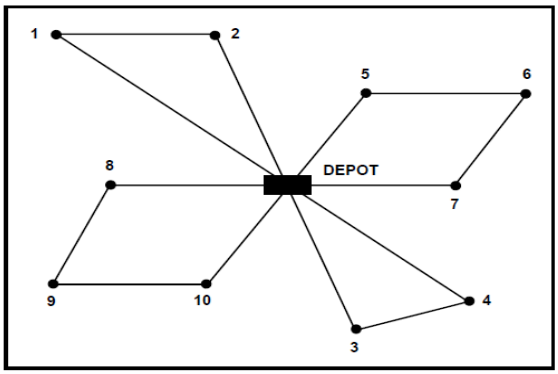
ภาพที่ 2.3 ผลลัพธ์การจัดเส้นทางเดินรถขนส่งสินค้าแบบเส้นทางเดียว (STSP)

2.5.2 การจัดเส้นทางเดินรถขนส่งสินค้าแบบหลายเส้นทาง (The Multiple Traveling Salesman Problem: MTSP) เป็นปัญหาที่มีความซับซ้อนมากยิ่งขึ้นซึ่งการจัดลำดับการขนส่งสินค้าแบบใช้เส้นทางหลายเส้นทางเพื่อขนส่งสินค้าให้กับลูกค้าตามจุดต่าง ๆ โดยจะเดินทางออกจากศูนย์กระจายสินค้าเดียว ไม่มีข้อจำกัดทางด้านเวลาและขนาดความจุบรรทุกของรถ ซึ่งผลลัพธ์ของเส้นทางที่จัดได้ ดังตัวอย่างแสดงในภาพที่ 2.4



ภาพที่ 2.4 ผลลัพธ์การจัดเส้นทางเดินรถขนส่งสินค้าแบบหลายเส้นทาง (MTSP)

2.5.3 การจัดเส้นทางเดินรถขนส่งสินค้าแบบคลาสสิค (The Classical Vehicle Routing Problem: CVRP) ปัญหาในระดับนี้เป็นการหาจำนวนเส้นทาง และการจัดลำดับการขนส่งสินค้าให้กับลูกค้าตามจุดต่างๆ โดยจะเดินทางออกจากศูนย์กระจายสินค้าเดียวซึ่งรับทราบความต้องการสินค้าของลูกค้าแต่ละรายภายใต้ข้อจำกัดทางด้านเวลาและขนาดความจุบรรทุกของรถที่กำหนดไว้ ดังตัวอย่างแสดงในภาพที่ 2.5



ภาพที่ 2.5 ผลลัพธ์การจัดเส้นทางเดินรถขนส่งสินค้าแบบคลาสสิค (CVRP)

2.6 เทคนิคการหาคำตอบ

2.6.1 วิธีหาคำตอบโดยใช้แบบจำลองทางคณิตศาสตร์ เป็นการนำเอาปัญหาที่เกิดขึ้นจริงมาสร้างเป็นสมการแบบจำลองขึ้น โดยจะมีการสร้างแบบจำลองในส่วนที่เป็นสมการ วัตถุประสงค์และส่วนที่เป็นข้อจำกัดเพื่อนำไปใช้ในการแก้ปัญหาซึ่งสามารถใช้คนหรือคอมพิวเตอร์ในการแก้ไขปัญหาได้ แต่ในการใช้คนกรณีที่ปัญหาที่มีตัวแปรมากอาจจะทำให้ไม่สามารถหาคำตอบที่ดีได้ดังนั้นการใช้คอมพิวเตอร์ช่วยนั้นจะทำให้สามารถรองรับปัญหาที่ใหญ่ได้และใช้เวลาในการหาคำตอบเร็วกว่าการใช้คน อย่างไรก็ตามการหาคำตอบโดยใช้วิธีการทางคอมพิวเตอร์จะได้คำตอบที่ใกล้เคียงกับคำตอบที่เหมาะสมที่สุดแต่ยังใช้เวลาในการหาคำตอบที่นานถ้าตัวแปรเยอะมากเวลาในการหาคำตอบจะทวีคูณขึ้นไปตัวอย่างของวิธีการแก้ปัญหาโดยใช้แบบจำลองทางคณิตศาสตร์ ของการจำลองเส้นทางแบบปัญหา VRP ดังนี้

กำหนด

n = จำนวนตำแหน่ง k = จำนวนรถบรรทุก (k =1, *f, NV*)

พารามิเตอร์

NV = จำนวนของรถบรรทุก C_k = ความจุของรถบรรทุก k

Q_i = ปริมาณ ความต้องการ ที่ จุด *I* d_{ij} = ระยะทาง ที่สั้นที่สุด จากจุด i ถึง j

p_{ij} = เวลาเมื่อถึงจุด *I* t_{ij} = เวลาขนส่งสินค้าจากจุด i ถึง j

T_k = เวลาสูงสุดที่อนุญาต สาหรับรถบรรทุก k

ตัวแปร

X^K_{ij} = รถบรรทุก k เดินทางจากจุด i ถึง j X^K_{ip} = รถบรรทุก k เดินทางจากจุด i ถึง p

X^K_{pj} = รถบรรทุก k เดินทางจากจุด p ถึง j

X = *matrix* X_{ij} = $\sum_{k=1}^{nv} x^k{}_{ij}$ กำหนดการเชื่อมต่อของเส้นทาง

U_i , U_j = ตัวสร้างขึ้นเพื่อป้องกันไม่ให้ Sub-tour ที่จุด i และ j ตามลำดับ

S= $\{(X_{ij}) : \sum_{i \in q} \sum_{i \in q} x_{ij} \geq 1,\}$ สำหรับชุด Q ที่เหมาะสม

ฟังก์ชันวัตถุประสงค์อาจจะเป็นอย่างใดอย่างหนึ่งลดระยะการขนส่งรวมหรือลดผลรวมของระยะทางที่ขนส่งทั้งหมดและเวลาการขนส่งทั้งหมดกรณีแรก ; ลดระยะทางที่ขนส่งทั้งหมด

Minimize

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{nv} d_{ij} x_{ij}^k \quad (1.1)$$

Subject To

$$\sum_{j=1}^{n} \sum_{k=1}^{nv} x_{ij}^k = 1 \qquad j = 2,\ldots\ldots, n$$

$$(1.2) \sum_{k=1}^{nv} x_{ij}^k = 1 \qquad i = 2,\ldots\ldots, n$$

$$(1.3) \sum_{i=1}^{n} x_{ip}^k - \sum_{k=1}^{nv} x_{ij}^k = 0 \qquad p = 1,\ldots\ldots, n \text{ ; } k=1,\ldots\ldots,NV$$

$$(1.4) \sum_{i=1}^{n} \sum_{j=1}^{n} Q_i x_{ij}^k < C_k \qquad k = 1,\ldots\ldots,NV$$

$$(1.5) \sum_{i=1}^{n} \sum_{j=1}^{n} p_i x_{ij}^k + \sum_{i=1}^{n} \sum_{j=1}^{n} t_i x_{ij}^k < T_k \qquad k = 1,\ldots\ldots,NV$$

(1.6) $\sum_{i=2}^{nv} x_{1j}^{k} < 1$ k = 1,.....,*NV*

(1.7) $\sum_{i=2}^{nv} x_{i1}^{k} < 1$ k = 1,.....,*NV*

(1.8) $X_{ij}^{k} = 0$ k = 1,.....,*NV* ; i = j where { $_{j=1}^{i=1}$

(1.9) U I = U j + Sx_{ij}^{k} < S − 1 i ≠ *j* ; *i* = 2 .....*n* ; *j* = 2 ,....,*n* ; *k* = 1,....,*NV*

(1.10) $\forall X_{ij}^{k} \in \{0,1\}$

เงื่อนไขที่ 1.1 ต้นทุนการเดินทางจากจุด i ไปจุด j โดยยานพาหนะ K โดยใช้ค่าใช้จ่ายน้อยที่สุด เงื่อนไขที่ 1.2 เงื่อนไขประกันว่าจุด j ใดๆ จะได้รับการเดินทางผ่านโดยยานพาหนะใดๆ อย่างน้อย 1 ครั้งโดยใช้เส้นทางที่ผ่านมาจากจุด i ใดๆ เงื่อนไขที่ 1.2 และ 1.3 กำหนดว่ารถขนส่งสินค้าสามารถเข้าถึงแต่ละจุดที่ต้องส่งสินค้าให้กับลูกค้าแต่ละรอบ เงื่อนไขที่ 1.4 เงื่อนไขการส่งสินค้าให้กับลูกค้ารายหนึ่ง ๆ จะเดินทางเข้าและออกเท่ากัน (1 ครั้ง) เงื่อนไขที่ 1.5 เงื่อนไขประกันว่ายานพาหนะใด ๆ จะขนส่งสินค้าไปส่งให้กับลูกค้าไม่เกินจำนวนที่สามารถบรรทุกได้ เงื่อนไขที่ 1.6 เงื่อนไขที่แสดงให้เห็นว่า ผลรวมของเวลาในการประมวลผลทั้งหมดและเวลาในการขนส่งทั้งหมดไม่เกินเวลาที่ได้รับอนุญาตให้รถแต่ละคันในแต่ละรอบ เงื่อนไขที่ 1.7 เงื่อนไขที่แสดงให้เห็นว่ารถบรรทุกหนึ่งคันสามารถออกจากศูนย์กระจายสินค้าได้เพียงครั้งเดียว เงื่อนไขที่ 1.8 เงื่อนไขที่แสดงให้เห็นว่า รถบรรทุกสามารถส่งของไปยังลูกค้า ได้เพียงครั้งเดียว เงื่อนไขที่ 1.7 และ 1.8 เงื่อนไขที่แสดงให้เห็นว่ารถบรรทุกสามารถออกและส่งของให้กับลูกค้า เพียงครั้งเดียวในแต่ละรอบ เงื่อนไขที่ 1.9 เงื่อนไขเพื่อป้องกันไม่ให้รถบรรทุกไปส่งสินค้ายังจุดเดิมเงื่อนไขที่ 1.10 เงื่อนไขเพื่อป้องกันไม่ให้เกิดทัวร์ย่อย (Sub tour) เงื่อนไขที่ 1.11 เงื่อนไขที่แสดงให้เห็นว่า ตัวแปรทั้งหมด X สามารถเป็น 0 หรือ 1

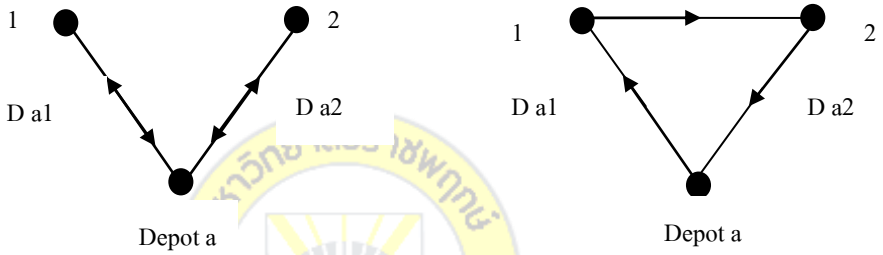
2.6.2 วิธีการหาคำตอบโดยใช้วิธีฮิวริสติก เป็นวิธีการที่อาศัยการกำหนดกฎเกณฑ์บางประการขึ้นมาโดยใช้สามัญสำนึกของมนุษย์เข้าช่วยในการแก้ปัญหา เพื่อหาคำตอบที่ดีและเหมาะสมในระดับหนึ่งถึงแม้ไม่ใช่คำตอบที่ดีที่สุด แต่สามารถให้คำตอบได้ภายในเวลาที่เหมาะสมไม่นานเกินไป รูปแบบการแก้ไขปัญหาและการค้นหาคำตอบเป็นวิธีการที่คิดอย่างมีเหตุผลตามประสบการณ์และความสมเหตุสมผลที่มีการพัฒนามานานกว่า 20 ปี วิธีการแก้ไขปัญหาจะถูกพัฒนาการตามระดับความยากง่าย ของปัญหาโดยมีการนำสามัญสำนึกของมนุษย์ผนวกเข้ากับแบบจำลองทางคณิตศาสตร์ เทคนิคการหาคำตอบด้วยวิธีฮิวริสติกส์เป็นแนวทางที่พยายามลดความซับซ้อนของปัญหาดังนั้นวิธีฮิวริสติกส์จึงมีรูปแบบในการแก้ปัญหาที่ค่อนข้างยืดหยุ่นอย่างมาก ส่งผลให้ในปัญหาหนึ่งๆอาจมีวิธีในการแก้ปัญหาแบบฮิวริสติกส์ที่แตกต่างๆกันมากมายหลายวิธีและแม้ว่าวิธีฮิวริสติกส์จะได้รับผล

เฉลยที่ดีที่สุด แต่วิธีนี้ก็มีจุดเด่นอยู่ที่ความรวดเร็วในการคำนวณผลเฉลย เทคนิคการแก้ปัญหาด้วยวิธีฮิวริสติกส์สามารถแบ่งวิธีการวิเคราะห์เพื่อหาคำตอบได้ 2 กลุ่ม คือ

1) กลุ่ม Classical heuristics
2) กลุ่ม Metaheuristics

เทคนิคที่นิยมใช้สำหรับการจัดเส้นทางเดินรถกลุ่ม Classical heuristics

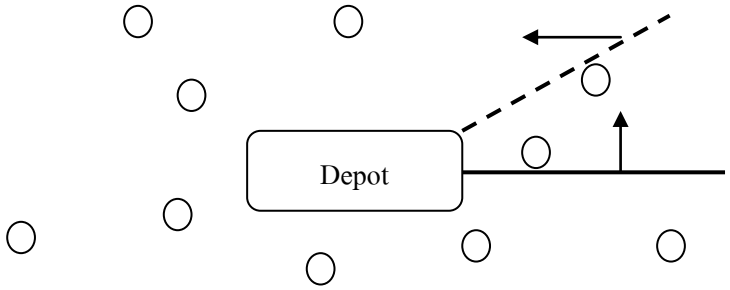
1) Saving หรือ Insertion Procedure เป็นวิธีการรวมจุดส่งสินค้าจุดอื่นๆเข้าไว้ในเส้นทางหลักแทนการจัดส่งสินค้าเป็น 2 เส้นทาง การรวมจุดส่งทำให้เกิดการประหยัดในการเดินทาง ค่าการประหยัด Saving จากจุดส่งสินค้าจุดที่ 2 มารวมกับเส้นทางหลักของลูกค้าจุดที่ 1 ดังแสดงในภาพที่ 2.6



ภาพที่ 2.6 แสดงวิธีการรวมจุดส่งสินค้าโดยวิธี saving: Clark and Wright saving

ทั้งนี้ค่าการประหยัดอาจพิจารณาจากค่าขนส่ง ระยะทางหรือค่าอื่นๆ ตามแต่ความเหมาะสม อีกทั้งควรพิจารณาความเป็นไปได้ของเส้นทางใหม่ในกรณีที่มีการส่งสินค้าด้วยวิธีการรวมจุดส่งว่าสามารถจัดส่งสินค้าได้ตามสภาพความเป็นจริงหรือไม่

2) Sweep Approach เป็นวิธีการหาจำนวนเส้นทางและลำดับการส่งสินค้าโดยการแบ่งเส้นทางเป็นพื้นที่รับผิดชอบด้วยการหมุนเส้นสมมติในทิศทางทวนเข็มนาฬิกาและรวมปริมาณสินค้าที่เส้นดังกล่าวผ่านร้านต่างๆ จนกระทั่งหมุนเส้นครบรอบ หลังจากนั้นจึงใช้เทคนิคการแก้ปัญหา TSP สำหรับแต่ละคัน แต่อย่างไรก็ตามวิธีการนี้อาจมีข้อบกพร่อง คือ ถ้าศูนย์กลางค้าไม่ได้อยู่ ณ ศูนย์กลางของพื้นที่ให้บริการ จะทำให้รูปร่างของเส้นทางการเดินรถมีขนาดไม่สมดุล และทำให้ระยะเวลาในการเดินทางของรถแต่ละคันมากกว่าวิธีการอื่นๆ นอกจากนี้แล้ววิธีการแบบนี้ไม่ได้คำนึงถึงลักษณะการวางแนวของถนนทำให้ร้านค้าที่อยู่ในถนนเดียวกัน อาจจะอยู่คนละเส้นทางก็ได้ ซึ่งไม่ถูกต้องกับความเป็นจริง



ภาพที่ 2.7 การแก้ปัญหาโดยวิธี Sweep Approach

3) Cluster Fist – Route Second เป็นการหาเส้นทางในการเดินรถโดยแบ่งพื้นที่ความรับผิดชอบในการส่งสินค้าก่อน หลังจากนั้นจึงหาลำดับในการส่งสินค้าในลพดับต่อไป

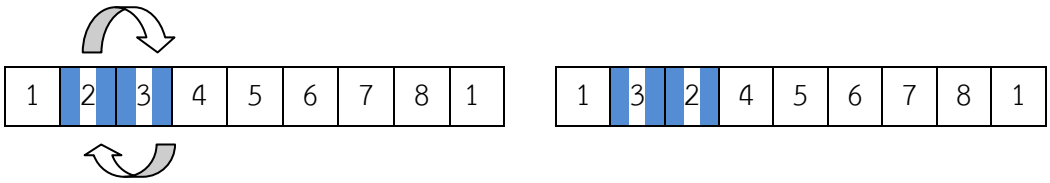
4) Route First Second วิธีการนี้จะเริ่มจากการหาเส้นทางเดินรถที่เหมาะสมที่ผ่านลูกค้าทุกราย ซึ่งเรียกเส้นทางนี้ว่า Giant Tour หลังจากนั้นจึงแบ่งออกเป็นเส้นทางย่อย ๆ เนื่องจากรถเพียงคันเดียวไม่สามารถเดินทางได้ครบทุกจุด ดังนั้นจึงต้องแบ่งเป็นเส้นทางย่อยโดยคำนึงถึงความจุของรถและข้อจำกัดระยะทางเวลาการขนส่ง

5) Improvement หรือ Exchange Procedure เป็นวิธีการในการปรับปรุงเส้นทางที่มีอยู่ด้วยการหาเส้นทางที่ช่วยลดค่าใช้จ่ายในการขนส่งจากการแลกเปลี่ยนเส้นทางที่มีอยู่แล้ว จนกระทั่งไม่สามารถปรับปรุงเส้นทางได้

6) Nearest Neighbor เป็นวิธีการในการจัดแบ่งคลาส เทคนิคนี้จะตัดสินใจ ว่าคลาสใดที่จะแทนเงื่อนไขหรือกรณีใหม่ ๆ ได้บ้าง โดยการตรวจสอบจำนวนบางจำนวน (“K” ใน K-nearest neighbor) ของกรณีหรือเงื่อนไขที่เหมือนกันหรือใกล้เคียงกันมากที่สุด โดยจะหาผลรวม (Count Up) ของจำนวนเงื่อนไข หรือกรณีต่างๆสำหรับแต่ละคลาส และกำหนดเงื่อนไขใหม่ๆ ให้คลาสที่เหมือนกันกับคลาสที่ใกล้เคียงกันมากที่สุดในการนำเทคนิคของ K-NN ไปใช้ในนั้นเป็นการหาวิธีการวัดระยะห่างระหว่างแต่ละ Attribute ในข้อมูลให้ได้ และจากนั้นคำนวณค่าออกมา ซึ่งวิธีนี้จะเหมาะสำหรับข้อมูลแบบตัวเลข แต่ตัวแปรที่เป็นค่าแบบไม่ต่อเนื่องนั้นก็สามารถทำได้ เพียงแต่ต้องการการจัดการแบบพิเศษเพิ่มขึ้น อย่างเช่น ถ้าเป็นเรื่องของสีจะใช้อะไรวัดความแตกต่างระหว่างสีน้ำเงินกับสีเขียว ต่อจากนั้นเราต้องมีวิธีในการรวมค่าระยะห่างของ Attribute ทุกค่าที่วัดมาได้ เมื่อเราสามารถคำนวณระยะห่างระหว่างเงื่อนไขหรือกรณีต่าง ๆ ได้จากนั้นเราเลือกชุดของเงื่อนไขที่ใช้จัดคลาสมาเป็นฐานสำหรับการจัดคลาสในเงื่อนไขใหม่ๆ ได้แล้วเราจะตัดสินได้ว่าขอบเขตของจุดข้างเคียงที่ควรเป็นนั้น ควรมีขนาดใหญ่เท่าไร และอาจตัดสินใจได้ด้วยว่าจะนับจำนวนจุดข้างเคียงตัวมันได้อย่างไร

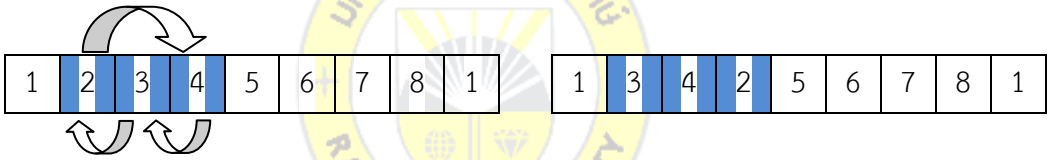
เทคนิคที่นิยมใช้สำหรับการจัดเส้นทางเดินรถกลุ่ม Metaheuristics

1) เทคนิคการปรับปรุงเส้นทางด้วยวิธี 2 – OPT เป็นเทคนิคการปรับปรุงเส้นทางภายในเส้นทางเดียวไม่มีการปรับปรุงระหว่างเส้นทางโดยวิธีการปรับปรุงคือการสลับลำดับจุดส่งสินค้าซึ่งในการเลือกสลับลำดับจุดส่งนั้นจะถูกเลือกสลับครั้งละ 2 จุดส่ง เพื่อทำการเปรียบเทียบผลลัพธ์การจัดเส้นทางระหว่างก่อนปรับปรุงและหลังปรับปรุงเพื่อเลือกเส้นทางที่ให้ผลลัพธ์ที่ดีที่สุดดังตัวภาพที่ 2.8



ภาพที่ 2.8 เทคนิคการปรับปรุงเส้นทางด้วยวิธี 2 – OPT

2) เทคนิคการปรับปรุงเส้นทางด้วยวิธี 3 – OPT เป็นเทคนิคการปรับปรุงเส้นทางเดียวที่มีความคล้ายคลึงกับเทคนิคการปรับปรุงเส้นทางด้วยวิธี 2 – OPT แต่การสลับลำดับจุดส่งนั้นจะถูกเลือกสลับครั้งละ 3 จุด วิธีนี้จะเกิดการสลับลำดับครั้งละ 3 จุด เพื่อให้เกิดเส้นทางใหม่ที่ให้ผลลัพธ์ของการจัดเส้นทางที่ดีขึ้น แสดงดังภาพที่ 2.9



ภาพที่ 2.9 เทคนิคการปรับปรุงเส้นทางด้วยวิธี 3 – OPT

3) เทคนิคการปรับปรุงเส้นทางด้วยวิธี Genetic Algorithm เป็นวิธีการค้นหาและแก้ปัญหาที่เหมาะสมที่สุด โดยพัฒนาและจำลองมาจากกระบวนการทางพันธุกรรม คือ ทฤษฎีการอยู่รอดของสิ่งมีชีวิต Genetic Algorithm เป็นทางเลือกที่ใช้แก้ปัญหาที่ดีเพราะวิธีการใช้แก้ปัญหาเป็นวิธีการแบบสุ่มและช่วยแก้ปัญหาที่มีขนาดใหญ่และซับซ้อน ได้เนื่องจากมีคุณสมบัติการเลียนแบบการถ่ายทอดทางพันธุกรรมตามธรรมชาติ ซึ่งจะนำค่าที่เหมาะสมที่สุดจากประชากรรุ่นก่อนมาใช้ในการพิจารณาในการหาคำตอบของประชาชนรุ่นถัดมาซึ่งจะถ่ายทอดลักษณะทางพันธุกรรรมจากบรรพบุรุษไปสู่รุ่นลูกหลานโดยใช้ค่า Fitness Function ที่สอดคล้องกับ Objective Function ในการพิจารณาหาคำตอบโดยมีการพิจารณาว่า โครโมโซมใดควรที่จะนำมาสืบพันธ์ หรือไม่ควรนำมาสืบพันธ์ จะทำให้สามารถหาคำตอบที่มีค่าสูงสุดหรือต่ำสุดที่สมบูรณ์ได้โดยการถ่ายทอดลักษณะทางพันธุกรรมจะมีกระบวนการที่ทำให้เกิดการเปลี่ยนแปลงที่เรียกว่า วิวัฒนาการ นั่นคือกระบวนการ Selection , Crossover , Mutation โดยคำนวณหาค่าความเหมาะสม Fitness Function

ที่สอดคล้องกับวัตถุประสงค์ของปัญหา Objective Function กำหนดให้กับโครโมโซมแต่ละตัว นำไปสู่กระบวนการคัดเลือก แต่บางปัญหาไม่สามารถที่จะคำนวณหาค่าเพื่อให้ได้ค่าความเหมาะสม ที่ตรงกับความจริงที่โครโมโซมนั้นควรจะได้รับและสิ่งนี้จึงทำให้เกิดปัญหาขึ้น ดังนั้นในการกำหนดค่าความเหมาะสมของโครโมโซมแต่ละตัว จึงเป็นสิ่งที่สำคัญ

4) เทคนิคการปรับปรุงเส้นทางด้วยวิธีค้นหาต้องห้าม Tabu Search เป็นวิธีการค้นหาคำตอบแบบเมตาฮิวริสติก (Meta-Heuristics) ซึ่งเป็นวิธีที่นิยมมากในการนำมาแก้ปัญหาที่ไม่เป็นโพลีเมียล (NP-Problem) เช่นปัญหาการกำหนดเส้นทางการขนส่งรถบรรทุก หรือปัญหาการจัดตารางการผลิต เป็นการยากที่จะหาคำตอบที่เหมาะสมที่สุด (OPTimal Solution) โดยเฉพาะอย่างยิ่งเมื่อปัญหานั้นมีขนาดใหญ่ (NP-hard) การหาคำตอบที่เหมาะสมที่สุดอาจใช้เวลาในการคำนวณมากหรือเป็นไปไม่ได้ที่หาคำตอบที่เหมาะสมที่สุด วิธีการค้นหาคำตอบแบบเมตาฮิวริสติกจึงถูกนำมาใช้ในการแก้ไขปัญหาเพราะใช้เวลาในการคำนวณน้อยกว่ามาก อีกทั้งคำตอบที่ได้จากวิธีเมตาฮิวริสติกก็สามารถยอมรับได้ในการนำไปใช้งานจริงดังนั้นการใช้วิธีการค้นหาคำตอบแบบเมตาฮิวริสติก (Meta-Heuristics) หลาย ๆ วิธีได้ถูกนำมาประยุกต์ใช้ในการแก้ไขปัญหาต่างๆอย่างแพร่หลาย

5) เทคนิคการปรับปรุงเส้นทางด้วยวิธีการอาณานิคมมด Ant Colony OPTimization เป็นวิธีที่ได้แรงบัลดาลใจมากจากพฤติกรรมของมดที่ออกไปหาอาหาร โดยมดจะเดินทางจากรังไปสู่แหล่งอาหารและกลับที่รังอีกครั้งหลังจากได้อาหารแล้วในระหว่างเดินทาง มดจะปล่อยสารเคมีที่มีชื่อว่าฟีโรโมน เพื่อให้มดตัวอื่นๆเดินทางตามกลิ่นหรือร่องรอยของฟีโรโมนนั้น ๆ เมื่อระยะเวลาผ่านไปฟีโรโมนนี้สามารถจะระเหยไปได้ตามคุณสมบัติทางเคมี ดังนั้นระยะทางที่ยาวเกินไปจะทำให้ฟีโรโมนระเหยหมดระหว่างเดินทาง ในขณะเดียวกันหากระยะทางสั้นจะทำให้มีโอกาสเพิ่มฟีโรโมนในขณะเดินทางไปกลับได้ก่อนที่ฟีโรโมนจะระเหย เมื่อมีการเดินทางของมดหลายๆตัวหรือมีการเดินทางหลายๆรอบ จะทำให้ฟีโรโมนเพิ่มความเข้มข้นขึ้นเรื่อยๆในทางตรงข้ามกับเส้นทางเดินที่มีระยะยาวกว่า จะมีแรงดึงดูดหรือระดับความเข้มข้นของฟีโรโมนมีความเจือจางและไม่ดึงดูดมดตัวอื่น ๆ ในที่สุด

6) เทคนิคการปรับปรุงเส้นทางด้วยวิธีการจำลองการอบเหนียว Simulated Annealing เป็นทฤษฎีที่จำลองมาจากกลศาสตร์สถิติ ซึ่งเป็นการศึกษาถึงลักษณะโครงสร้างทางกายภาพโดยรวมของสสาร ซึ่งจะมีอะตอมประกอบอยู่เป็นจำนวนถึง 10^{23} หน่วยต่อลูกบาศก์เซนติเมตร โดยลักษณะพฤติกรรมต่างๆ จะแสดงออกมาตามแรงกระตุ้น และการเปลี่ยนแปลงอุณหภูมิที่ได้รับจึงเปรียบเสมือนเป็นการหาค่าเฉลี่ยของผลที่เกิดขึ้นจากทุก ๆ อะตอมในระบบโดยรวบรวมจากค่าสถิติที่เกิดนั่นเอง ซึ่งปัญหากลศาสตร์สถิติในขั้นตอนสุดท้ายจะมีการศึกษาถึงผลกระทบต่อระบบขณะที่อุณหภูมิเข้าสู่สถานะพื้นฐาน ซึ่งเป็นช่วงที่พลังงานของระบบมีค่าต่ำสุด โดยกระบวนการแอนนิลลิ่ง Annealing จะประกอบไปด้วยขั้นตอนต่างๆดังนี้

1. ทำการหลอมระบบ (Melting) ระบบในครั้งแรกด้วยอุณหภูมิที่สูง
2. ทำให้อุณหภูมิลดต่ำลงอย่างช้าๆ โดยลดตามตารางอุณหภูมิ Annealing
3. ขณะที่อุณหภูมิใกล้เคียงกับจุดเยือกแข็ง ให้คงสภาพนั้นไว้ในช่วงระยะหนึ่ง

โดยในแต่ละช่วงของการลดอุณหภูมิให้ต่ำลงนี้กระบวนการ Annealing จะต้องจัดสรรเวลาที่นานพอที่จะทำให้ระบบเกิดภาวะสมดุล หรือเรียกว่าภาวะเสถียรและเมื่อระบบเกิดภาวะเสถียรในแต่ละช่วงอุณหภูมิจนกระทั่งพบว่าไม่มีการพัฒนาผลลัพธ์ที่ได้อีกแล้ว ซึ่งแสดงถึงระบบได้เข้าสู่สถานะพื้นฐานแล้วระบบก็จะทำการหยุดลง

Metropolis, et.al. 1953 ได้แนะนำวิธี มอนติคาร์โล เพื่อใช้ในการศึกษาพฤติกรรมของอะตอม ขณะเข้าสู่สภาวะสมดุลที่อุณหภูมิต่างๆ ซึ่งมีขั้นตอนดังนี้

1. กำหนดรูปร่างของส่วนประกอบของกลศาสตร์สถิติ
2. การย้ายตำแหน่งของส่วนประกอบต่างๆ และคำนวณหาผลของการเปลี่ยนแปลงของพลังงาน
3. ผลการเปลี่ยนแปลงพลังงาน $\wedge E$ หรือ δ ที่เกิดจาก $E_{old} - E_{new}$ น้อยกว่าศูนย์แสดงว่าเกิดการเปลี่ยนแปลงพลังงานที่เพิ่มขึ้นเป็นสิ่งที่ดีต่อระบบ ดังนั้นให้ยอมรับการย้ายตำแหน่งของอะตอมที่เกิดขึ้น และนำไปเป็นรูปร่างของระบบใหม่ในขั้นตอนต่อไป แต่หากพบว่าผลการเปลี่ยนแปลงพลังงาน $\wedge E$ มากกว่าหรือเท่ากับศูนย์แสดงถึงการเปลี่ยนแปลงพลังงานที่ลดลงในทิศทางที่ไม่ดี จะยอมรับการย้ายตำแหน่งที่เกิดขึ้นด้วยความน่าจะเป็น $P \wedge E = e^{(- \wedge E : KbT)}$ โดย

e แทน ฟังก์ชันเอ็กโปเนนเชียล

T แทน อุณหภูมิ

Kb แทน ค่าคงที่ของโบลส์แมน

วิธีมอนติคาร์โล วิธีการนี้จะมีการสุ่มตัวอย่างเพื่อใช้ในการจำลองสถานการณ์ที่เกี่ยวข้องกับการเปรียบเทียบค่า $P \wedge E$ ด้วยตัวเลขสุ่มที่แจกแจงแบบยูนิฟอร์มในช่วง (0,1) ทำให้มีโอกาสในการยอมรับการย้ายตำแหน่งของอะตอมที่เกิดขึ้นในระบบ และทำให้พลังงานของระบบเพิ่มขึ้น แทนการปฏิเสธการย้ายตำแหน่งของอะตอมนั้น

กระบวนการต่างๆ เหล่านี้จะดำเนินการไปเพื่อให้ระบบเข้าสู่สภาวะสมดุล ณ อุณหภูมิปัจจุบันจากนั้นจะทำการลดอุณหภูมิลงตามตารางอุณหภูมิแอนนิลลิ่งจนกระทั่งระบบถึงสถานะพื้นฐาน ซึ่งมีพลังงานในระบบต่ำ

Kirkpatrick และคณะ ได้กล่าวถึงซิมูเลทแอนนิลลิ่งไว้ว่าเป็นการหาคำตอบโดยการประมาณซึ่งเกิดจากการนำสองทฤษฎีมาเชื่อมกันระหว่าง ทฤษฎีทางกายภาพในการอบอ่อนของวัสดุแข็ง รวมกับทฤษฎีทางการหาคำตอบที่ดีที่สุดแบบการรวม โดยซิมูเลทแอนนิลลิ่งจะเริ่มต้นจากการกลอม

ระบบในครั้งแรกด้วยอุณหภูมิที่สูงจากนั้นจะทำการลดอุณหภูมิลงอย่างช้าๆ จนกระทั่งจุดเยือกแข็งและไม่เกิดการเปลี่ยนแปลงใดๆ ขึ้นอีกภายในระบบ

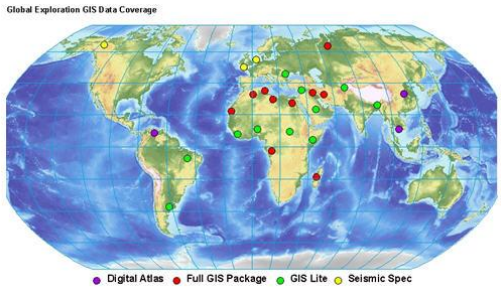
ในแต่ละช่วงอุณหภูมิที่มีการเปลี่ยนแปลงนี้ กระบวนการจะต้องจัดสรรเวลาที่นานพอที่จะให้ระบบเกิดสภาวะเสถียร ซึ่งลักษณะการเกิดสถานะเสถียรจะประยุกต์มาจากการหาคำตอบที่ดีที่สุดด้วยวิธีการรวม ที่ได้กล่าวไว้ในขั้นต้น คือ ระบบการจัดเรียงรูปแบบใหม่ จนกระทั่งพบว่าการจัดเรียงใหม่นี้มีการพัฒนาที่ดีขึ้น ระบบก็จะนำการจัดเรียงใหม่ที่ได้ดังกล่าวมาเป็นจุดเริ่มต้นในการดำเนินการใหม่ ซึ่งระบบจะดำเนินการอย่างต่อเนื่องจนกระทั่งพบว่าไม่มีการพัฒนาผลลัพธ์ที่ได้แล้วระบบก็หยุดตัวลง

ขั้นตอนของ Simulated Annealing

ขั้นตอนของการหาค่าที่ดีที่น้อยที่สุด Simulated Annealing จะเริ่มต้นจากการสร้างผลลัพธ์โดยอาจจะเกิดจากการสุ่มตัวอย่างขึ้นมาหรือผ่านกระบวนการทางเลือกทางฮิวริสติกส์ทั่วไป จากนั้นระบบจะพิจารณาถึงการเปลี่ยนแปลงพลังงานภายในระบบ $\wedge E$ ($E_{old} - E_{new}$) หรือ δ ซึ่งข้อกำหนดของ $\wedge E$ (δ) $\wedge E$ >=0 แสดงถึงพลังงานใหม่มีน้อยกว่าของเดิมส่งผลการเปลี่ยนแปลงพลังงานลดลง ระบบจะยอมรับว่าได้คำตอบที่ดีกว่าดังนั้นผลลัพธ์ก็จะถูกใช้เป็นจุดเริ่มต้นในขั้นตอนถัดไป แต่ถ้า $\wedge E$ < 0 แสดงถึงพลังงานในระบบเพิ่มขึ้นหรือหมายถึงคำตอบก่อนหน้ามีค่ามากกว่าครั้งก่อนหน้า วิธีการทางความน่าจะเป็นจะเข้ามาเกี่ยวข้องในการประเมินโดยระบบจะทำการสุ่มตัวเลขที่มีการแจกแจงแบบยูนิฟอร์ม ขึ้นมาเปรียบเทียบ $P_{\wedge E}$ ซึ่ง $P_{\wedge E}$ จะมีค่าเท่ากับ $e^{(-\wedge E : KbT)}$ หรือ $e^{(\delta : KbT)}$ ซึ่งถูกเรียกอีกนัยหนึ่งว่าการแจกแจงแบบโบล์สแมน ถ้ากรณีตัวเลขที่สุ่มขึ้นมาน้อยกว่า $P_{\wedge E}$ ผลลัพธ์เดิมก็จะถูกนำไปใช้เป็นจุดเริ่มต้นในขั้นตอนถัดไป ระบบจะทำการดำเนินการจนกระทั่งไม่พบคำตอบที่ดีไปกว่าที่หาได้ระบบก็จะทำการหยุดลง

2.7 ระบบสารสนเทศทางภูมิศาสตร์

ระบบสารสนเทศทางภูมิศาสตร์ คือ ระบบการทางานที่ผสมผสานกันระหว่างฮาร์ตแวร์ ซอฟแวร์ และข้อมูลต่างๆ เช่น ข้อมูลที่ได้จากการถ่ายภาพ หรือภาพถ่ายดาวเทียมรวมถึงการรวบรวม การบริหารจัดการ การวิเคราะห์ และการแสดงรูปแบบข้อมูลทางแผนที่ต่าง ๆ เข้าด้วยกัน ซึ่งจะช่วยให้มองเห็น และเข้าใจภาพรวมทั้งหมดที่แสดงออกมาผ่านภาพ แผนที่ และรายงานต่างๆ ได้อย่างทั่วถึง ไม่เพียงเท่านี้ระบบ GIS ยังสามารถตอบคำถามที่ต้องการได้อีกด้วย ทั้งนี้ระบบ GIS สามารถนำมาผสมผสานและประยุกต์ใช้ได้กับทุกหน่วยงาน ซึ่งนับว่ามีอรรถประโยชน์มากหากนำใช้ให้ถูกวิธีนั่นเอง



ภาพที่ 2.10 ระบบสารสนเทศทางภูมิศาสตร์

การเกิดขึ้นครั้งแรกของ GIS ได้ถือกำเนิดขึ้นในปี ค.ศ. 1960 ณ แถบอเมริกาเหนือ โดยหน่วยงาน Canada Geographic Information System ในประเทศแคนนาดา ซึ่งเหตุผลในการจัดตั้ง GIS ขึ้นนั้น เนื่องจากแคนาดาต้องการรายได้หลักทางเศรษฐกิจ ซึ่งต้องอาศัยทรัพยากรทางธรรมชาติเป็นจำนวนมาก ภาระหน้าที่หลักของหน่วยงานดังกล่าว คือ การสำรวจเพื่อการวางแผนดำเนินการ เกี่ยวกับทรัพยากรป่าไม้ แร่ธรรมชาติ แหล่งที่อยู่อาศัย แหล่งน้ำ โดยการทางานในระยะเริ่มต้นยังมีข้อจำกัดทางด้านเทคนิคหลายประการ เช่น

- ข้อมูลที่มีปริมาณมากเกินไป
- คอมพิวเตอร์ที่มีกำลังและประสิทธิภาพไม่เพียงพอ
- เครื่องพิมพ์สามารถพิมพ์ได้แค่อักษร และเส้นตรงเท่านั้น
- ป้อนข้อมูลทางด้าน Graphic ไม่ได้
- ระบบไม่สามารถทำการวิเคราะห์ข้อมูลได้

จนกระทั่งในช่วง ค.ศ. 1980-1990 GIS มีการพัฒนามากขึ้น เนื่องจากเริ่มมีการเล็งเห็นถึงความสำคัญของ GIS กันมากขึ้น รวมทั้งมีปัจจัยความก้าวหน้าทางพัฒนาการคอมพิวเตอร์ และระบบการเชื่อมโยงเครือข่ายข้อมูล (Networking) การพัฒนาทางเทคโนโลยีทางคอมพิวเตอร์นี้เอง จึงทำให้การจัดเก็บข้อมูลในระบบสามารถจุได้มากขึ้น สามารถมองภาพรวมและองค์ประกอบต่างๆ ของพื้นที่ในภาพรวมได้ง่ายขึ้น

นอกจาก GIS จะสามารถผลิตแผนที่ได้อย่างสวยงามและมีการแสดงผลผ่านจอแสดงผลแล้ว GIS ยังสามารถสอบถามข้อมูลและวิเคราะห์ข้อมูลผ่านแผนที่บนระบบคอมพิวเตอร์ รวมทั้งสามารถเรียกค้นข้อมูลมาดูได้หลายข้อมูลพร้อมกันจากการแสดงผลเป็นชั้น ข้อมูล (Layer) ทำให้ง่ายต่อการวิเคราะห์ และประมวลผลที่มีการพัฒนาเรื่อยมาจนถึงปัจจุบัน

GIS มาจากคำว่า GEOGRAPHIC INFORMATION SYSTEM หรือแปลเป็นภาษาไทยว่า ระบบสารสนเทศภูมิศาสตร์ ซึ่งหมายถึง กระบวนการทางานเกี่ยวกับการจัดเก็บข้อมูลในเชิงพื้นที่

(spatial data) ด้วยระบบคอมพิวเตอร์โดยการกำหนดข้อมูลเชิงบรรยาย (attribute data) เพื่อนำมาประมวลผล หรือวิเคราะห์ทำแบบจำลองต่าง ๆ และแสดงผลในรูปของข้อมูลเชิงพื้นที่ เพื่อนำมาใช้สนับสนุนในการตัดสินใจแก้ปัญหา การวางแผนที่ซับซ้อน เช่น ที่อยู่ บ้านเลขที่ ที่มีความสัมพันธ์กับตำแหน่งในเชิงพื้นที่ เช่น ตำแหน่ง เส้นรุ้ง เส้นแวง ในรูปของตารางข้อมูล และฐานข้อมูล หรือจะเรียกให้เข้าใจง่าย ๆ ว่า smart map นั่นเอง

ระบบสารสนเทศภูมิศาสตร์ เป็นโปรแกรมที่สามารถนำไปใช้ในการสร้างและวิเคราะห์ข้อมูลรูปทรงสัณฐานของวัตถุทุกอย่างบนพื้นผิวโลก (Spatial) เกี่ยวกับระบบแผนที่ภาพถ่ายทางอากาศและแผนผังต่างๆ ของลักษณะภูมิประเทศทั้งที่เกิดขึ้นเองตามธรรมชาติและที่มนุษย์สร้างขึ้น สิ่งเหล่านี้สามารถแปลและถอดออกมาเป็นรหัสอิเล็กโทรนิกส์ ซึ่งสามารถเรียกออกมาใช้งาน เพื่อแก้ไขและวิเคราะห์ข้อมูลได้ แต่จากการสำรวจอัตราส่วนในการนำไปใช้ประโยชน์ถือว่า ประสบผลสำเร็จค่อนข้างสูงมากในปัจจุบัน เพราะมีพัฒนาการที่เจริญก้าวหน้าทางเทคโนโลยี อีกทั้งยังมีการนำมาประยุกต์ใช้ให้เข้ากับความต้องการและการทางานของแต่ละหน่วยงาน

ในกระบวนการการทางานของ GIS จำเป็นต้องใช้คอมพิวเตอร์ฮาร์ดแวร์ (Hardware) ซอฟต์แวร์ (Software) ข้อมูลทางภูมิศาสตร์ (Geographic Data) และการออกแบบ (Personnel Design) ในการเสริมสร้างประสิทธิภาพของการจัดเก็บข้อมูล การปรับปรุงข้อมูล การคำนวณ และการวิเคราะห์ข้อมูล ให้แสดงผลในรูปของข้อมูลที่สามารถอ้างอิงได้ในทางภูมิศาสตร์ กล่าวง่าย ๆ คือการใช้สมรรถนะของคอมพิวเตอร์ในการจัดเก็บและการใช้ข้อมูลเพื่ออธิบายสภาพต่างๆ บนพื้นผิวโลกโดยอาศัยลักษณะทางภูมิศาสตร์ เป็นตัวเชื่อมโยงความสัมพันธ์ระหว่างข้อมูลต่างๆนั้นเอง เนื่องจาก GIS ใช้กล้องดาวเทียมในการทางาน จึงมีส่วนช่วยให้การทางานของ smart map มีประสิทธิภาพในการทางานสูง สามารถนำไปประยุกต์ใช้ได้กับทุกหน่วยงาน เช่น ช่วยให้ตรวจสอบถึงลักษณะของพื้นที่นั้นๆ ได้อย่างละเอียด ในมุมมองที่กว้างขึ้นอีกด้วย หรือการแสดงภาพเหตุการณ์นั้น ๆ ในปัจจุบัน ดังนั้นจะเห็นได้ว่า GIS นั้นมีประสิทธิภาพในการทางานมาก สามารถนำมาประยุกต์ใช้และเป็นประโยชน์ต่อหลายหน่วยงานอีกด้วย เช่น การตรวจสอบพื้นที่ที่ยากต่อการเข้าถึง นอกจากจะช่วยลดค่าใช้จ่ายในการเดินทาง หรือแรงงานในการทางานแล้ว smart map ยังช่วยเพิ่มความถูกต้องแม่นยำ ไม่เพียงเท่านี้ ระบบ GIS ยังมีประโยชน์ต่อการวางผังเมือง ช่วยในการวางแผนล่วงหน้า ทั้งระบบไฟฟ้า ประปา ท่อน้ำทิ้ง บ่อบำบัดน้ำเสีย ฯลฯ และมีการแสดงผลที่ง่ายต่อการทำความเข้าใจอีกด้วย

องค์ประกอบในการทางานของระบบ GIS ในการทางานของระบบ GIS จะมีองค์ประกอบต่าง ๆ ดังนี้

ภาพที่ 2.11 องค์ประกอบในการทางานของระบบ GIS

1. อุปกรณ์คอมพิวเตอร์ (hardware) หมายถึง คอมพิวเตอร์และเครื่องต่อพ่วงอื่น ๆ ทั้งเครื่องพิมพ์ สแกนเนอร์ กล้องถ่ายรูป ที่มีหน้าที่ในการนำเข้าข้อมูล ประมวลผล แสดงผล และผลิตผลลัพธ์ของการทางาน

2. โปรแกรม (software) คือ กลุ่มโปรแกรมสำเร็จรูปที่ติดตั้งบนระบบฮาร์ดแวร์ เพื่อให้ระบบสารสนเทศภูมิศาสตร์ สามารถทางานได้ตามที่ได้รับการออกแบบไว้ โดยมีโปรแกรมหลัก คือ โปรแกรม WINDOW, UNIX โปรแกรมระบบสารสนเทศภูมิศาสตร์ เช่น โปรแกรม ARC/INFO โปรแกรม PAMAP โปรแกรม INTERGRAPH, AutoCAD MAP, MAPINFO นอกจากนั้นอาจมีโปรแกรมช่วยงานต่าง ๆ (Utilities) เช่น โปรแกรมช่วยจัดการหน่วยความจำ โปรแกรม Editor อีกด้วย ซึ่งโปรแกรมต่าง ๆ เหล่านี้ จะประกอบด้วยฟังก์ชัน การทางานและเครื่องมือที่จำเป็นต่าง ๆ สำหรับนำเข้าและปรับแต่งข้อมูล จัดการระบบฐานข้อมูล เรียกค้น วิเคราะห์ และจำลองภาพนั่นเอง

3. ขั้นตอนการทางาน (Methods) คือ วิธีการที่องค์กรนั้น ๆ นำเอาระบบ GIS ไปใช้งานโดยแต่ละ ระบบ แต่ละองค์กรย่อมมีความแตกต่างกันออกไป ฉะนั้น ผู้ปฏิบัติงานต้องเลือกวิธีการในการจัดการกับปัญหาที่เหมาะสมที่สุด สำหรับของหน่วยงานนั้น ๆ เอง ซึ่งความสำเร็จของการใช้ระบบจะขึ้นอยู่กับแผนงานออกแบบ และการกำหนดขั้นตอนการปฏิบัติงาน เพื่อให้งานเป็นไปตามขั้นตอนและมีความน่าเชื่อถือ ซึ่งจะแตกต่างไปตามความเหมาะสมของชนิดงานนั่นเอง

4. บุคลากร (people) คือ บุคคลที่มีความรู้พื้นฐานทางด้านคอมพิวเตอร์ และทางด้านภูมิศาสตร์ สามารถวิเคราะห์ และออกแบบแผนที่และแผนภูมิที่เป็นผลลัพธ์ของการวิเคราะห์ เพื่อแสดงผลได้อย่างถูกต้องตามมาตรฐานว่าด้วยวิชาการออกแบบแผนที่ (Cartography) โดยจำแนกบุคลากรตามลักษณะงานดังนี้ เช่น พนักงานภาคสนาม พนักงานเตรียมข้อมูลและต้นร่าง พนักงาน

ป้อนข้อมูล พนักงานวิเคราะห์ข้อมูล และพนักงานออกแบบแผนที่ เป็นต้น ทั้งนี้บุคลากรที่เกี่ยวข้องกับระบบสารสนเทศภูมิศาสตร์ ถือเป็นองค์ประกอบที่สำคัญที่สุดในระบบ GIS เนื่องจากถ้าขาดบุคลากรเหล่านี้ ข้อมูลที่มีอยู่มากมายมหาศาลนั้น ก็จะเป็นเพียงขยะไม่มีคุณค่าใดเลย เพราะไม่ได้ถูกนำไปใช้งานนั่นเอง

5. ข้อมูล (data) แหล่งข้อมูลของระบบสารสนเทศภูมิศาสตร์ที่สำคัญ ได้แก่ แผนที่ภูมิประเทศ มาตราส่วน 1:50,000 รูปถ่ายทางอากาศ (Aerial Photographs) หรือ ภาพถ่ายดาวเทียม (Satellite Imagery) นอกเหนือจากข้อมูลเชิงพื้นที่ข้างต้นแล้ว ระบบสารสนเทศยังต้องการข้อมูลเชิงบรรยาย ที่จะช่วยขยายความด้านรายละเอียดของข้อมูลเชิงพื้นที่ ตัวอย่างของข้อมูลเชิงบรรยาย ได้แก่ ชื่อของหมู่บ้าน จำนวนครัวเรือน จำนวนประชากรชาย-หญิง เป็นต้น แหล่งที่มาของข้อมูลเชิงบรรยายอาจได้มาจากหน่วยงานที่เกี่ยวข้องหรือได้มา จากการสำรวจข้อมูลภาคสนาม (Field Data Collection) ก็ได้ ข้อมูลเชิงบรรยายจะถูกบันทึกเก็บในลักษณะของบันทึก (Record) โดยแต่ละบันทึกจะถูกแบ่งย่อยออกเป็นช่องสนาม (Field) ช่องสนามแต่ละช่องอาจถูกกำหนดให้บันทึกข้อมูลที่เป็นตัวอักษร (Alphabetic) หรือ ข้อมูลที่เป็นตัวเลข (Numeric) ก็แล้วแต่ความเหมาะสมไม่เพียงเท่านี้ ข้อมูลต่างๆ ที่ใช้ในระบบ GIS นั้นจะถูกจัดเก็บในรูปแบบของฐานข้อมูล โดยได้รับการดูแลจากระบบจัดการฐานข้อมูลหรือ DBMS ข้อมูลถือเป็นองค์ประกอบที่สำคัญรองลงมาจากบุคลากร

ขั้นตอนการทางานของ GIS

ในส่วนของขั้นตอนการทางาน (Methods) คือ วิธีการที่องค์กรนั้น ๆ นำเอาระบบ GIS ไปใช้งาน ซึ่งจะแตกต่างไปตามแต่ละองค์กร ฉะนั้นผู้ปฏิบัติงานต้องเลือกวิธีการในการจัดการกับปัญหาที่เหมาะสมที่สุดสำหรับของหน่วยงานของตนเอง ทั้งนี้ก็เพื่อประโยชน์และประสิทธิภาพที่สูงสุดของหน่วยงานนั่นเอง

1. การนำเข้าข้อมูล (input) ก่อนที่ข้อมูลทางภูมิศาสตร์จะถูกใช้งานได้ในระบบสารสนเทศภูมิศาสตร์ ข้อมูลจะต้องถูกแปลงให้อยู่ในรูปของข้อมูลเชิงตัวเลข (digital format) ก่อน เช่น จากแผนที่กระดาษไปสู่ข้อมูลในรูปแบบดิจิตอลหรือแฟ้มข้อมูลบนเครื่อง คอมพิวเตอร์อุปกรณ์ที่ใช้ในการนำเข้าเช่น Digitizer Scanner หรือ Keyboard เป็นต้น

2. การปรับแต่งข้อมูล (manipulation) ข้อมูลที่ได้รับเข้าสู่ระบบ บางอย่างจำเป็นต้องได้รับการปรับแต่งให้เหมาะสมกับการใช้งาน เช่น ข้อมูลบางอย่างมีขนาด หรือสเกล (scale) ที่แตกต่างกัน หรือใช้ระบบพิกัดแผนที่ที่แตกต่างกัน ข้อมูลเหล่านี้จะต้องได้รับการปรับให้อยู่ในระดับเดียวกันเสียก่อน

3. การบริหารข้อมูล (management) ระบบจัดการฐานข้อมูลหรือ DBMS จะถูกนำมาใช้ในการบริหารข้อมูลเพื่อการทางานที่มีประสิทธิภาพในระบบ GIS DBMS ที่ได้รับการเชื่อถือและนิยมใช้

กันอย่างกว้างขวางที่สุดคือ DBMS แบบ Relational หรือระบบจัดการฐานข้อมูลแบบสัมพัทธ์ (DBMS) ซึ่งมีหลักการทางานพื้นฐาน ดังนี้คือ ข้อมูลจะถูกจัดเก็บในรูปของตารางหลาย ๆ ตาราง

4. การเรียกค้นและวิเคราะห์ข้อมูล (query and analysis) เมื่อระบบ GIS มีความพร้อมในเรื่องของข้อมูลแล้ว ขั้นตอนต่อไป คือ การนำข้อมูลเหล่านี่มาใช้ให้เกิดประโยชน์ เช่น ใครคือเจ้าของกรรมสิทธิ์ในที่ดินผืนที่ติดกับโรงเรียน? ต้องมีการสอบถามอย่างง่าย ๆ เช่น ชี้เมาส์ไปในบริเวณที่ต้องการแล้วเลือก (Point and click) เพื่อสอบถามหรือเรียกค้นข้อมูล นอกจากนี้ระบบ GIS ยังมีเครื่องมือในการวิเคราะห์ เช่น การวิเคราะห์เชิงประมาณค่า (proximity หรือ buffer) การวิเคราะห์เชิงซ้อน (overlay analysis) เป็นต้น

5. การนำเสนอข้อมูล (visualization) จากการดำเนินการเรียกค้นและวิเคราะห์ข้อมูล ผลลัพธ์ที่ได้จะอยู่ในรูปของตัวเลขหรือตัวอักษร ซึ่งยากต่อการตีความหมายหรือทำความเข้าใจ การนำเสนอข้อมูลที่ดี เช่น การแสดงชาร์ต (chart) แบบ 2 มิติ หรือ 3 มิติ รูปภาพจากสถานที่จริง ภาพเคลื่อนไหว แผนที่ หรือแม้กระทั่งระบบมัลติมีเดียสื่อต่างๆ เหล่านี้จะทำให้ผู้ใช้เข้าใจความหมายและมองภาพของผลลัพธ์ที่กำลังนำเสนอได้ดียิ่งขึ้นอีกนั่นเอง

จากที่กล่าวมาทั้งหมดข้างต้นนั้น แสดงให้เห็นถึงความหลากหลายในการนำมาใช้งานของระบบ GIS ซึ่งหลากหลายและมีประโยชน์รอบด้านมาก ซึ่งจะแบ่งออกเป็นหัวข้อใหญ่ๆ ดังนี้

1. การอนุรักษ์และจัดการสิ่งแวดล้อม (Environmental Management, Conservation) การจัดการทางพืชและสัตว์ในดิน (Flora and Fauna) สัตว์ป่า (Wild Life) อุทยานแห่งชาติ (National Park) การควบคุมและติดตามมลภาวะ (Pollution Control and Monitoring) และแบบจำลองด้านนิเวศวิทยา (Ecological Modeling)

2. การจัดการด้านทรัพยากร/การเกษตร (Resources Management / Agriculture) การจัดการระบบชลประทานทำการพัฒนาและจัดการที่ดินเพื่อการเกษตร การอนุรักษ์ดินและน้ำ การจัดการทรัพยากรธรรมชาติ ป่าไม้ และการทำไม้

3. การวางแผนด้านสาธารณะภัย (Disaster Planning) การบรรเทาสาธารณะภัย การติดตามการปนเปื้อนของสารพิษ และแบบจำลองผลกระทบอุทกภัย (Modeling Flood Impacts)

4. ด้านผังเมือง (Urban GIS) การวางแผนผังเมือง การใช้ประโยชน์ที่ดินสิ่งปลูกสร้างต่าง ๆ เช่น ถนน เขื่อน คลอง เป็นต้น การตรวจสอบการเคลื่อนย้ายถิ่นฐานของประชากร การเปลี่ยนแปลงของการใช้พื้นที่ การวางผังเมือง การวิเคราะห์ด้านอาชญากรรม ที่ดินและภาษีที่ดิน ระบบการระบายน้ำเสีย โครงการพัฒนาที่อยู่อาศัย ฯลฯ

5. การจัดการสาธารณูปโภค (Facilities Management) การจัดการด้านไฟฟ้า ประปา ท่อส่งก๊าซ หน่วยดับเพลิง ระบบจราจรและโทรคมนาคม

6. การวิเคราะห์ด้านตลาด (Marketing Analysis) การหาที่ตั้ง ที่เหมาะสมในการขยายสาขาหรือสำนักงาน

7. ด้านการเดินทาง แสดงแผนที่ เส้นทาง จุดสำคัญในการเดินทาง เป็นประโยชน์ต่อการท่องเที่ยว

8. ด้านประโยชน์ทางการทหาร

9. ด้านสาธารณสุข เพื่อดูการแพร่ขยายของโรคระบาดต่างๆ เช่น ไข้หวัด

10. ด้านโบราณคดี

จากประโยชน์ต่างๆข้างต้นจะพบว่า ในปัจจุบันกิจกรรมต่างๆ ล้วนมีความจำเป็นที่จะต้องใช้ประโยชน์จากแผนที่ และระบบ GIS ก็เข้ามามีบทบาทสำคัญเพื่อประกอบในกานตัดสินใจ จึงอาจกล่าวได้ว่า GIS เป็นเครื่องมือสำคัญสำหรับองค์กร ตลอดจนผู้บริหารและจ้าหน้าที่ทุกๆ ส่วน ซึ่งสามารถมองภาพรวมของแผนที่ต่างๆ และลักษณะองค์ประกอบของพื้นที่ได้อย่างชัดเจน จึงง่ายต่อการตัดสินใจวางโครงการ หรือประกอบธุรกรรมต่างๆ ได้เป็นอย่างดี

ในปัจจุบัน GIS ได้รับความนิยมอย่างแพร่หลาย เพราะเป็น IT ที่มีประสิทธิภาพในการทางานสูง ช่วยให้สามารถเห็นภาพหรือพื้นที่ต่าง ๆ ได้ในมุมกว้างอย่างชัดเจน อีกทั้งสะดวกต่อการทางานและช่วยให้การวิเคราะห์พื้นที่ต่าง ๆ เป็นไปอย่างง่ายดาย เพราะไม่จำเป็นต้องออกเดินทางไปยังพื้นที่นั้นๆด้วยตนเอง หรือช่วยอำนวยความสะดวกในกรณีที่เราต้องออกเดินทางไปในที่ๆ ยากต่อการเข้าถึงและมีข้อจำกัดด้านเวลา หรือระยะทางในการเดินทาง เป็นต้น ดังนั้น GIS จึงช่วยลดต้นทุนในการเดินทางได้เป็นอย่างดี

พัฒนาการของเทคโนโลยีในปัจจุบัน มีส่วนช่วยส่งเสริมให้การดำเนินงานต่าง ๆ ไปเป็นอย่างสะดวกสบายมากขึ้น จึงไม่แปลกที่ GIS จะได้รับความนิยมและสามารถนำมาประยุกต์ใช้ให้เข้ากับความต้องการของแต่ละองค์กร ด้วยเหตุนี้เอง GIS จึงเป็นที่แพร่หลายมากในประเทศที่พัฒนาแล้ว อีกทั้งยังถูกพัฒนาความสามารถให้ดีและมีประสิทธิภาพมากขึ้นเรื่อย ๆ อย่างไม่หยุดยั้ง

ดังนั้น ในอนาคตคาดว่า GIS คงมีระบบและขั้นตอนการทำงานที่ง่ายและสะดวกสบายกับผู้ที่ใช้งานมากขึ้น เพราะเทคโนโลยีจะมีความเจริญก้าวหน้าอย่างต่อเนื่อง และได้รับการพัฒนาให้เป็นประโยชน์กับผู้ใช้โดยตรง มีตัวเลือกที่ช่วยในการวิเคราะห์ทางาน หรือตอบคำถามที่เราสงสัยได้มากกว่านี้ สามารถแสดงภาพและลักษณะของพื้นที่ดังกล่าวได้ใกล้เคียงความเป็นจริง มีราคาต้นทุนการใช้ที่ถูกลง และยังสามารถนำมาประยุกต์ใช้งานได้อย่างแพร่หลายกับทุก ๆ หน่วยงานอีกด้วย

2.8 Visual Basic

เป็นเครื่องมือในการสร้างแอพพลิเคชั่น (โปรแกรมประยุกต์) ที่ทำงานภายใต้ระบบปฏิบัติการ Windows ซึ่งแอพพลิเคชั่นที่ได้จะมีรูปแบบการติดต่อกับผู้ใช้เป็นกราฟิก (GUI: Graphic User Interface) วิชวลเบสิก จะใช้ Event (เหตุการณ์) เป็นตัวขับเคลื่อนคำสั่ง (event-driven) หมายความว่า โค้ดของโปรแกรม จะยังไม่ทำอะไร (idle) จนกว่ามันจะถูกเรียกให้ตอบสนองต่อ event (เช่น การคลิกเมาส์ การกดปุ่ม การเลือกเมนู ฯลฯ) นั่นคือจะไม่มีอะไรเกิดขึ้นจนกว่า event จะถูกตรวจพบเมื่อนั้นแหละ code ของโปรแกรมก็จะตอบสนองต่อ event นั้น ๆ ในลักษณะของโปรแกรมย่อย ๆ หรือโมดุล (event procedure)

2.9 กรณีศึกษา บริษัทผู้ให้บริการขนส่งทางด้านโลจิสติกส์

กรณีศึกษาเป็นบริษัทขนส่งสินค้าจากโรงงานไปยังลูกค้าปลายทาง ได้เปิดให้บริการกระจายสินค้าไปยังโมเดิร์นเทรดโดยเริ่มเป็นผู้ให้บริการกระจายสินค้าให้กับ บริษัท บิ๊กซี ซุปเปอร์เซ็นเตอร์ จำกัด (มหาชน) และ บริษัท สยามแม็คโคร จำกัด (มหาชน) นับเป็นจุดเริ่มแรกของการนำระบบกระจายสินค้าแบบ Cross dock มาใช้ในงานด้านการขนส่งเป็นต้นมาจนถึงปัจจุบัน หลังจากนั้นในปี พ.ศ. 2552 ได้มีการขยายธุรกิจการขนส่งสินค้าเข้าร้านค้าโชห่วยทั่ว เพื่อเพิ่มศักยภาพในการกระจายสินค้า เข้าร้านค้า 20 จังหวัด ทั่วภาคอีสานให้แข็งแกร่งยิ่งขึ้น นอกจากนี้ยังขยายงานด้านการบริหารคลังสินค้า อย่างเต็มรูปแบบเพื่อเป็นผู้ให้บริการด้านโลจิสติกส์แบบครบวงจร การบริการของบริษัทกรณีศึกษามีดังนี้

- บริการกระจายสินค้าในรูปแบบ Cross-Docking เข้าโมเดิร์นเทรด (Makro , Big C , Big C-Extra,Tesco Lotus)

- บริการขนส่งสินค้าจากจุดหนึ่งไปยังอีกจุดหนึ่ง (Door to Door) เช่น การขนส่งสินค้าระหว่างโรงงาน-คลังสินค้า, โรงงาน-โรงงาน เป็นต้น โดยบรรทุกสินค้าเต็มคันรถ

- บริการกระจายสินค้าไปยังร้านค้า (Traditional Trade) ทั่วประเทศ เช่น HomePro , VILLA MARKET , TOPS Supermarket, FOODLAND, Big C Market, Big C Mini ,ไทวัสดุ และห้างฯท้องถิ่น ทั่วประเทศ

- บริการธุรกิจ E-Commerce ทั้งในด้านการให้บริการระบบคลังสินค้า, แพ็คสินค้า และระบบขนส่ง รวมถึงระบบสารสนเทศ เพื่อลดค่าใช้จ่าย และเพิ่มศักยภาพ ทางการแข่งขันให้กับ ผู้ขายสินค้า

- บริการจัดส่งสินค้า และพัสดุด่วนพิเศษ (Express) ที่ครอบคลุมทั่วประเทศ เป็นต้น

2.10 งานวิจัยที่เกี่ยวข้อง

ทวินันท์ สิมะจารึก และคณะ (2552) ได้นำเสนองานวิจัยเรื่องการลดค่าใช้จ่ายในการขนส่ง กรณีศึกษาโรงงานเคมีภัณฑ์ เพื่อวิเคราะห์หาสาเหตุของปัญหาในการขนส่งและหาแนวทางการลดค่าใช้จ่ายการขนส่งสินค้า พบว่าค่าเชื้อเพลิงเป็นปัญหาที่ทำให้ต้นทุนของบริษัทสูง ซึ่งมีสาเหตุมาจากปัจจัยภายใน คือ การกำหนดเส้นทางที่เป็นมาตรฐานและปัจจัยภายนอก คือ อัตราค่าเชื้อเพลิงในตลาดโลกที่มีความผันผวน คณะผู้วิจัยจึงได้กำหนดเส้นทางการขนส่งสินค้าแบบใหม่ โดยใช้เทคนิคการแก้ปัญหาเส้นทางแบบวิธีการแบบจำลองการขนส่ง (Transportation model) และวิธีการแก้ปัญหาการจัดเส้นทางสำหรับรถบรรทุก (Truck routing) รวบรวมสินค้าเพื่อจัดเส้นทางให้ลูกค้าที่มีเส้นทางการขนส่งในทางเดียวกันรวมเข้าด้วยกัน เพื่อลดจำนวนเที่ยวและระยะทางในการขนส่งสินค้า หลังจากนั้นนำมาเปรียบเทียบหาวิธีการที่มีค่าใช้จ่ายต่ำสุดและเหมาะสมที่สุดจากผลการดำเนินงาน พบว่าวิธีการแบบจำลองการขนส่งและวิธีการแก้ปัญหาการจัดเส้นทางสำหรับรถบรรทุกทำให้จำนวนเที่ยว ระยะทางในการขนส่งสินค้า ค่าน้ำมันเชื้อเพลิง ค่าใช้จ่ายรวมทั้งหมด คิดเป็นเปอร์เซ็นต์ที่ลดลงจากการดำเนินงานแบบเดิม

วิไลวรรณ์ แก่นสาร และ สมบัติ สินธุเชาน์ (2556) ได้ศึกษาการเปรียบเทียบวิธีการฮิวริสติกส์ สำหรับระบบการจัดการขยะซึ่งจัดทำเพื่อแก้ปัญหาในการลดต้นทุนต่าง ๆ ทั้งทางตรงและทางอ้อม พร้อมทั้งเกิดการลดต้นทุนค่าใช้จ่ายได้เป็นอย่างดีอีกทั้งยังมีวิธีการที่เข้าใจง่ายไม่ซับซ้อนโดยเนื้อหาใน งานวิจัยนี้ได้บอกถึงการประยุกต์วิธีการหาคำตอบโดยวิธีการ Saving heuristic , Nearest heuristic และ Max-Nearest เพื่อทำการเปรียบเทียบกันโดยพิจารณาจากผลลัพธ์ที่ได้ภายใต้เงื่อนไขเดียวกัน โดยมีเงื่อนไขดังนี้ ความสามารถในการบรรจุที่เท่ากัน และ มีเวลาเข้ามาเกี่ยวข้อง ผลทดลองของ งานวิจัยนี้จะเห็นว่าวิธี Saving heuristic ให้คำตอบที่ดีที่สุดคือมีระยะทางรวม 55.022 กิโลเมตร และวิธี Nearest heuristic ต่อด้วยวิธี Max-nearest คือ 62.152 กิโลเมตร และ 67.196 กิโลเมตร ตามลำดับ ซึ่งในปัจจุบันเส้นทางในการเก็บขยะจะมีระยะทางรวมอยู่ที่ 62.246 กิโลเมตร ดังนั้นจากการทดลองจะเห็นว่าวิธี Saving และ Nearest สามารถลดระยะทางลงได้พร้อมกับสามารถควบคุมเวลาในการเก็บขยะได้อีกด้วย

อตินุช ลิมป์มณีรักษ์ และนระเกณฑ์ พุ่มชูศรี (2556) ได้ศึกษาการออกแบบระบบสารสนเทศสำหรับองค์กรกลางให้บริการการขนส่ง เพื่อนำเสนอหนทางที่สามารถลดความสูญเสียที่เกิดขึ้นในการขนส่ง โดยแบ่งการศึกษาวิจัยออกเป็น 3 ส่วน ได้แก่ 1.การนำเสนอแนวคิดระบบองค์กรกลางที่ไม่มีรถขนส่งเป็นของตนเองสำหรับบริการขนส่ง 2.การวิเคราะห์ระบบ 3.การออกแบบระบบสารสนเทศ ซึ่งในการเสนอแนวความคิดและการออกแบบระบบสารสนเทศนี้จะเป็นต้นแบบการจัดการรถที่มีความจุเหลือให้สามารถเกิดมูลค่าจากความจุที่สูญเสียเปล่า จากการศึกษาถึงการประสานความร่วมมือในการขนส่ง การวางแผนการขนส่งโดยจะช่วยให้สามารถใช้ประโยชน์จากความจุที่เหลือที่มีได้

สูงสุด และสามารถเป็นต้นแบบในการจัดการรถที่มีความจุเหลือให้สามารถเพิ่มมูลค่าจากความจุที่สูญเสียเปล่าอันนาไปสู่การลดต้นทุนการขนส่งได้

วัลลภ รัตนถาวร และ สีรง ปรีชานนท์ (2553) ได้ศึกษาฮิวริสติกสำหรับปัญหาตำแหน่งที่ตั้งศูนย์กระจายสินค้าแบบพลวัตโดยมีข้อจำกัดด้านขนาด เพื่อทำให้ค่าใช้จ่ายรวมของการตอบสนองต่อความต้องการของลูกค้าตลอดช่วงเวลาการวางแผนมีค่าน้อยที่สุด ด้วยการจัดตั้งศูนย์กระจายสินค้าที่เหมาะสม โดยจะพิจารณาจากความต้องการของลูกค้าว่ามีลักษณะเปลี่ยนแปลงไปตามเวลาและเป็นฟังก์ชันของเวลา งานวิจัยนี้ได้นำเสนอขั้นตอนวิธีการแก้ปัญหาด้วยวิธีการค้นหาแบบทำบู โดยใช้วิธีการประมาณเชตข้างเคียงแทนวิธีการหาคำตอบที่ดีที่สุด และวิธีการแก้ปัญหาแบบพันธุกรรม (GA) จากการทดสอบสมรรถนะของฮิวริสติกที่พัฒนาขึ้นสามารถสรุปได้ว่า วิธีการค้นหาแบบทำบูจะให้ประสิทธิภาพในด้านเวลาในการหาคำตอบที่ดีกว่าขั้นตอนวิธีพันธุกรรมในทุกชนิดของปัญหา โดยเฉพาะอย่างยิ่งเมื่อจำนวนคำบเวลาย่อยในระยะเวลาการวางแผนของปัญหาแบบพลวัตเพิ่มสูงขึ้น ด้านคุณภาพของคำตอบวิธีการค้นหาแบบทำบูจะให้คุณภาพของคำตอบที่ดีกว่าขั้นตอนพันธุกรรมเพียงเล็กน้อย

ไพโรจน์ แสนดี และคณะ (2557) ได้ศึกษาเส้นทางเดินรถในการเคลื่อนย้ายผู้ประสบอุทกภัยออกจากพื้นที่อันตรายเมื่อระดับน้ำสูง กรณีศึกษา : ตำบลลาดสวาย อำเภอลำลูกกา จังหวัดปทุมธานี เพื่อให้ได้เส้นทางการเดินรถที่มีระยะทางในการอพยพรวมต่ำที่สุด โดยพื้นที่กรณีศึกษาคัดเลือกมาจากบริเวณที่มีปัญหาน้ำท่วมรุนแรงในปี 2554 ได้แก่ ตำบลลาดสวาย อำเภอลาลูกกา จังหวัดปทุมธานี ซึ่งมีประชากร 46,921 คน โดยสัดส่วนผู้อพยพคิดเป็น 10% ของจำนวนประชากร แบ่งโซนในการอพยพออกเป็น 6 โซน ซึ่งการหาเส้นทางที่เป็นคำตอบได้ใช้วิธีการแบ่งพื้นที่ออกเป็นกลุ่มก่อนจัดเส้นทาง(Cluster – Firdt Route – Second) และวิธีการเปรียบเทียบการประหยัด(Saving Algorithm) ผลการศึกษาพบว่าวิธีการเปรียบเทียบการประหยัด (Saving Algorithm) ให้เส้นทางการอพยพ เป็นระยะทางรวม 909.39 กิโลเมตรและมีค่าใช้จ่ายรวมเท่ากับ 9,093.90 บาท ในขณะที่วิธีการแบ่งพื้นที่ออกเป็นกลุ่มก่อนจัดเส้นทาง (Cluster –Firdt Route – Second) ให้เส้นทางการอพยพ เป็นระยะทางรวม 1,442.73 กิโลเมตรและมีค่าใช้จ่ายรวมเท่ากับ 14,427.30 บาท โดยวิธีการเปรียบเทียบการประหยัด สามารถประหยัดค่าใช้จ่ายได้มากกว่าวิธีการแบ่งพื้นที่ออกเป็นกลุ่มก่อนจัดเส้นทาง ได้เท่ากับ 5,333.40 บาท

ทวินันท์ สิมะจารึก และคณะ (2552) ได้นำเสนองานวิจัยเรื่องการลดค่าใช้จ่ายในการขนส่ง กรณีศึกษาโรงงานเคมีภัณฑ์ เพื่อวิเคราะห์หาสาเหตุของปัญหาในการขนส่งและหาแนวทางการลดค่าใช้จ่ายการขนส่งสินค้า พบว่าค่าเชื้อเพลิงเป็นปัญหาที่ทำให้ต้นทุนของบริษัทสูง ซึ่งมีสาเหตุมาจากปัจจัยภายใน คือ การกำหนดเส้นทางที่เป็นมาตรฐานและปัจจัยภายนอก คือ อัตราค่าเชื้อเพลิงในตลาดโลกที่มีความผันผวน คณะผู้วิจัยจึงได้กำหนดเส้นทางการขนส่งสินค้าแบบใหม่ โดยใช้เทคนิคการ

แก้ปัญหาเส้นทางแบบวิธีการแบบจำลองการขนส่ง (Transportation model) และวิธีการแก้ปัญหาการจัดเส้นทางสำหรับรถบรรทุก (Truck routing) รวบรวมสินค้าเพื่อจัดเส้นทางให้ลูกค้าที่มีเส้นทางการขนส่งในทางเดียวกันรวมเข้าด้วยกัน เพื่อลดจำนวนเที่ยวและระยะทางในการขนส่งสินค้า หลังจากนั้นนำมาเปรียบเทียบหาวิธีการที่มีค่าใช้จ่ายต่ำสุดและเหมาะสมที่สุดจากผลการดำเนินงาน พบว่าวิธีการแบบจำลองการขนส่งและวิธีการแก้ปัญหาการจัดเส้นทางสำหรับรถบรรทุกทำให้จำนวนเที่ยว ระยะทางในการขนส่งสินค้า ค่าน้ำมันเชื้อเพลิง ค่าใช้จ่ายรวมทั้งหมด คิดเป็นเปอร์เซ็นต์ที่ลดลงจากการดำเนินงานแบบเดิม

อรประไพ จารุพัฒน์ และ ปวีณา เชาวลิตวงศ์ (2556) ได้ศึกษาฮิวริสติกส์สำหรับการจัดเส้นทางลำดับการส่งสินค้าของรถแต่ละคันจากจุดปล่อยรถแห่งเดียวหรือหลายแห่ง เพื่อให้ลูกค้าทุกรายได้รับสินค้าโดยปัญหาการจัดเส้นทางมีหลากหลายแตกต่างกันไปตามวัตถุประสงค์และข้อจำกัด สำหรับการจัดเส้นทางเดินรถตามลักษณะการเดินทางจะแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือ 1. เส้นทางเดินแบบปิด คือ รถแต่ละคันต้องออกและกลับมาสู่จุดปลายรถเดิม 2. เส้นทางเดินแบบเปิด คือ รถขนส่งไม่ต้องย้อนกลับมายังจุดปล่อยรถหลังจากส่งสินค้าให้กับลูกค้ารายสุดท้าย โดยได้นำวิธีฮิวริสติกส์มาใช้ในการจัดเส้นทางการขนส่ง คือ การจัดแบบแทรกงาน (Insertion Heuristic) ซึ่งการหาคำตอบของวิธีนี้ในแต่ละรอบจะทำการค้นหาว่า งานขนส่งที่รอการจัดลงเส้นทางในแต่ละจุดรับ-ส่งสามารถแทรกลงในเส้นทางที่จัดค้างอยู่ได้ตำแหน่งใดบ้าง โดยพยายามแทรกบนเส้นทางที่ทำให้เกิดค่าใช้จ่ายเพิ่มขึ้นน้อยที่สุด และต้องคำนึงถึงข้อจากัดหลักของความจุรถและเวลาในการส่งมอบ

พลอยพรรณ ศรีกิจการ และ อรอุไร แสงสว่าง (2556) ได้ศึกษาเส้นทางเดินรถขนส่งเครื่องสำอาง โดยจัดทำเพื่อปรับปรุงประสิทธิภาพของการจัดเส้นทางการเดินรถซึ่งได้ปรับปรุงเกี่ยวกับวางแผนและออกแบบเส้นทางเดินรถขนส่งเครื่องสำอางของบริษัทมาร์คอัพคอสเมติค โดยทำการจัดส่งสินค้าไปยังร้านตัวแทนจำหน่าย 20 ร้าน ในเขตกรุงเทพฯ และปริมณฑล ปัญหาของงานวิจัยนี้คือปัญหาการเดินทางของพนักงานขาย (Traveling Salesman problem) ซึ่งได้ประยุกต์ใช้หลักเมตาวิธีฮิวริสติกส์โดยวิธการจำลองการอบเหนียว (Simulated Annealing) ในการแก้ปัญหา เพื่อหาเส้นทางที่สั้นที่สุดในการเดินทางผ่านเมืองทุกๆเมือง เมืองละหนึ่งครั้งแล้วกลับมายังจุดเริ่มต้น จากวิธีการหาคำตอบที่ใกล้เคียงที่สุดซึ่งเป็นวิธีที่ใช้ในปัจจุบัน โดยส่งสินค้าไปยังจุดที่ใกล้ที่สุดก่อน มีระยะทางรวม 377.3 กิโลเมตรต่อวัน บริษัทจัดส่งสินค้าทุกวันศุกร์ที่ 2 และวันศุกร์ที่ 4 ของเดือน ในระยะเวลา 1 เดือน จะใช้ระยะทางการวิ่งงานรวมทั้งสิ้น 377.3 x 2 = 754.6 กิโลเมตร และเมื่อใช้วิธีการจำลองอบเหนียว (SA) ในการออกแบบเส้นทางเดินรถพบว่า รถสามารถวิ่งงานด้วยระยะทางเพียง 347.8 กิโลเมตรต่อวัน ดังนั้นในระยะเวลา 1 เดือน จะมีระยะทางรวมทั้งสิ้น 347.8 x 2 = 695.6 กิโลเมตร ซึ่งลดลงจากเดิม 7.81 เปอร์เซ็นต์ หรือ 59 กิโลเมตรต่อเดือน

อภิชิต มณีงาม และคณะ (2556) ได้ศึกษาการแก้ปัญหาการจัดเส้นทางการเดินรถโดยมีการจำกัดเวลาการเดินรถบรรทุกขนาดใหญ่ด้วยวิธีฮิวริสติกส์ ซึ่งได้นำเสนอวิธีฮิวริสติกส์สาหรับหาคำตอบของปัญหาการจัดเส้นทางการเดินในกรุงเทพมหานครและปริมณฑล โดยพิจารณาเงื่อนการจากัดเวลาเดินรถบรรทุกขนาดใหญ่ในเขตเมือง เพื่อให้ได้เส้นทางการเดินรถที่มีต้นทุนรวมต่ำที่สุด โดยได้พัฒนาวิธีการหาคำตอบเป็น 3 ขั้นตอนตามลำดับดังนี้ 1.จัดกลุ่มลูกค้าตามเงื่อนไขการจำกัดเวลาในการเดินรถ 2.สร้างคำตอบเริ่มต้นด้วยวิธีเซฟวิ่งอัลกอริทึม (Savings Algorithm) 3.นำคำตอบที่ได้จากขั้นตอนที่สองมาปรับปรุงเส้นทางโดยวิธีการปรับปรุงเฉพาะที่ ซึ่งใช้การแลกเปลี่ยนลูกค้าระหว่างเส้นทาง (Customer - exchange) ผสมกับ วิธีการย้ายลูกค้าหนึ่งรายระหว่างเส้นทาง (One-move Operator) ผลจากการศึกษาพบว่าการสร้างคำตอบเริ่มต้นด้วยวิธีเซฟวิ่งอัลกอริทึม และปรับปรุงคำตอบด้วยวิธีการแลกเปลี่ยนลูกค้าระหว่างเส้นทาง และผสมกับวิธีการย้ายลูกค้าหนึ่งรายระหว่างเส้นทาง ให้ผลการจัดเส้นทางที่มีระยะทางรวมต่ำที่สุดเมื่อเปรียบเทียบกับวิธีอื่นๆ ซึ่งวิธีการดังกล่าวสามารถลดระยะทางรวมจากเดิม 19,367 กิโลเมตร ลดลงเหลือ 17,489 กิโลเมตร หรือลดลง 9.70 เปอร์เซ็นต์ โดยใช้เวลาในการประมวลผลเพื่อหาคำตอบพร้อมกันทุกวิธีสูงสุดเพียง 9.72 วินาที อีกด้วย

อตินุช ลิมป์มณีรักษ์ และนระเกณฑ์ พุ่มชูศรี (2556) ได้ศึกษาการออกแบบระบบสารสนเทศสำหรับองค์กรกลางให้บริการการขนส่ง เพื่อนำเสนอหนทางที่สามารถลดความสูญเสียที่เกิดขึ้นในการขนส่ง โดยแบ่งการศึกษาวิจัยออกเป็น 3 ส่วน ได้แก่ 1.การนำเสนอแนวคิดระบบองค์กรกลางที่ไม่มีรถขนส่งเป็นของตนเองสำหรับบริการขนส่ง 2.การวิเคราะห์ระบบ 3.การออกแบบระบบสารสนเทศ ซึ่งในการเสนอแนวความคิดและการออกแบบระบบสารสนเทศนี้จะเป็นต้นแบบการจัดการรถที่มีความจุเหลือให้สามารถเกิดมูลค่าจากความจุที่สูญเสียเปล่า จากการศึกษาถึงการประสานความร่วมมือในการขนส่ง การวางแผนการขนส่งโดยจะช่วยให้สามารถใช้ประโยชน์จากความจุที่เหลือที่มีได้สูงสุด และสามารถเป็นต้นแบบในการจัดการรถที่มีความจุเหลือให้สามารถเพิ่มมูลค่าจากความจุที่สูญเสียเปล่าอันนำไปสู่การลดต้นทุนการขนส่งได้

ปาลีรัฐ บุญก่อน (2554) ได้ศึกษาการพัฒนาระบบการจัดเส้นทางขนส่งสินค้าจากศูนย์กระจายสินค้าไปยังลูกค้าตามจุดต่างๆ เพื่อช่วยสนับสนุนการตัดสินใจผู้ใช้ระบบ โดยมีศูนย์กระจายสินค้าแห่งเดียวและมีรูปแบบปัญหาการจัดเส้นทางการเดินรถแบบมีข้อจำกัดเรื่องความสามารถในการบรรทุกโดยพิจารณาเรื่องน้ำหนักและปริมาตรสินค้าให้อยู่ภายใต้เงื่อนไขที่จำกัดเรื่องความสามารถในการบรรทุกโดยพิจารณาเรื่องน้ำหนักและปริมาตรสินค้าให้อยู่ภายใต้เงื่อนไขของความจุรถโดยมีเป้าหมายเพื่อให้ได้ระยะทางที่สั้นที่สุด วิธีที่ใช้ในการจัดเส้นทางเดินรถขนส่งสินค้าของงานวิจัยนี้ คือ วิธีฮิวริสติกส์แบบ Saving Algorithm ของ Clarke and Wright โดยใช้วิธี 2 –OPT ปรับปรุงเส้นทางและมีการประยุกต์ใช้ระบบสารสนเทศทางภูมิศาสตร์ร่วมด้วยโดยจัดทำโปรแกรมลง

บน Visual Basic for Application (VBA) ใน Microsoft Excel ผลลัพธ์ที่แสดงออกมาสามารถระบุตำแหน่งจุดส่งสินค้า ปริมาตรและน้ำหนักสินค้าที่รถบรรทุกแต่ละคันบรรทุกไปซึ่งผลลัพธ์ที่ได้สามารถเพิ่มประสิทธิภาพอรรถประโยชน์การใช้รถมากขึ้นภายในระยะเวลาสั้นลงและสามารถลดเวลาที่ใช้ในการจัดรถได้ดีเมื่อเปรียบเทียบกับข้อมูลตัวอย่าง

วัลลภ รัตนถาวร และ สีรง ปรีชานนท์ (2553) ได้ศึกษาฮิวริสติกสำหรับปัญหาตำแหน่งที่ตั้งศูนย์กระจายสินค้าแบบพลวัตโดยมีข้อจำกัดด้านขนาด เพื่อทำให้ค่าใช้จ่ายรวมของการตอบสนองต่อความต้องการของลูกค้าตลอดช่วงเวลาการวางแผนมีค่าน้อยที่สุด ด้วยการจัดตั้งศูนย์กระจายสินค้าที่เหมาะสม โดยจะพิจารณาจากความต้องการของลูกค้าว่ามีลักษณะเปลี่ยนแปลงไปตามเวลาและเป็นฟังก์ชันของเวลา งานวิจัยนี้ได้นำเสนอขั้นตอนวิธีการแก้ปัญหาด้วยวิธีการค้นหาแบบทำบู โดยใช้วิธีการประมาณเซตข้างเคียงแทนวิธีการหาคำตอบที่ดีที่สุด และวิธีการแก้ปัญหาแบบพันธุกรรม (GA) จากการทดสอบสมรรถนะของฮิวริสติกที่พัฒนาขึ้นสามารถสรุปได้ว่า วิธีการค้นหาแบบทำบูจะให้ประสิทธิภาพในด้านเวลาในการหาคำตอบที่ดีกว่าขั้นตอนวิธีพันธุกรรมในทุกชนิดของปัญหาโดยเฉพาะอย่างยิ่งเมื่อจำนวนคำตอบเวลาย่อยในระยะเวลาการวางแผนของปัญหาแบบพลวัตเพิ่มสูงขึ้น ด้านคุณภาพของคำตอบวิธีการค้นหาแบบทำบูจะให้คุณภาพของคำตอบที่ดีกว่าขั้นตอนพันธุกรรมเพียงเล็กน้อย

มหาวิทยาลัยราชภัฏ RAJAPRUK UNIVERSITY

บทที่ 3
วิธีดำเนินการวิจัย

การวิจัยเรื่อง การจัดเส้นทางการขนส่งด้วยวิธีเมต้าฮิวริสติกส์ กรณีศึกษา บริษัทผู้ให้บริการขนส่งทางด้านโลจิสติกส์ ซึ่งการวิจัยนี้ผู้วิจัยได้กำหนด วิธีดำเนินการวิจัยมีขั้นตอนดังนี้

3.1 ประชากรที่ใช้ในการศึกษา

3.2 ขั้นตอนการดำเนินงานวิจัย

3.3 เครื่องมือที่ใช้ในงานวิจัย

3.1 ประชากรที่ใช้ในการศึกษา

กรณีศึกษาผู้วิจัยเลือกศึกษาแบบเจาะจง (Purposive Sampling) ใช้เพียงบริษัทเดียวเท่านั้น เนื่องจากบริษัทกรณีศึกษานี้ เป็นผู้ให้บริการขนส่งและกระจายสินค้าให้กับ ผู้ประกอบการอุตสาหกรรม ห้าง ร้าน ในเขตกรุงเทพมหานคร ปริมณฑล และต่างจังหวัด ปัจจุบันมีลูกค้าใช้บริการไม่น้อยกว่า 2,000 ราย และเป็นบริษัทที่ให้บริการด้านโลจิสติกส์แบบครบวงจร

3.2 ขั้นตอนการดำเนินงานวิจัย

ขั้นตอนการทำงานวิจัยประกอบด้วยขั้นตอนหลักๆ พอสรุปได้ดังนี้

1. ศึกษาขั้นตอนการทำงานของบริษัทกรณีศึกษา โดยเน้นศึกษาในส่วนการจัดเส้นทางขนส่งสินค้าพร้อมทั้งศึกษาแนวทางการพัฒนา
2. กำหนดปัญหาและวัตถุประสงค์ในการทำวิจัยครั้งนี้
3. ศึกษาค้นคว้าทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการจัดเส้นทางขนส่งสินค้าเพื่อนำมารวบรวมและใช้เป็นเอกสารประกอบการวิเคราะห์
4. วิเคราะห์ข้อมูลและเก็บรวบรวมข้อมูลที่เกี่ยวข้องในงานวิจัยครั้งนี้

จากการวิเคราะห์ข้อมูลที่ต้องใช้ในการพัฒนาสามารถแบ่งออกเป็นแต่ละประเภท ดังนี้

- ข้อมูลรถที่ใช้ในการขนส่งสินค้า เช่น ประเภทของรถ น้ำหนักที่รถแต่ละประเภทสามารถบรรทุกได้ ปริมาตรที่รถแต่ละประเภทสามารถบรรจุได้และจำนวนรถบรรทุกแต่ละประเภท
- ระยะเวลาที่พนักงานใช้ในการจัดเส้นทางการขนส่งสินค้า
- ข้อมูลเรื่องตำแหน่งที่ตั้งของลูกค้าแต่ละราย
- ขนาดและน้ำหนักของสินค้า
- ระยะทางจากคลังสินค้าไปยังลูกค้าแต่ละราย

5. ออกแบบฐานข้อมูลและค่าพารามิเตอร์ต่างๆที่ใช้ในการคำนวณทางโปรแกรม
6. ทำการทดลองการจัดเส้นทางขนส่งสินค้าด้วยโปรแกรม
7. สรุปผลในการทำงานวิจัยครั้งนี้

3.3 เครื่องมือที่ใช้ในงานวิจัย

งานวิจัยนี้ ผู้วิจัยได้สร้างเครื่องมือช่วยพนักงานในการจัดเส้นทางเดินรถโดยเครื่องมือนี้คือโปรแกรมที่ทำงานบน Microsoft Excel ร่วมกับ VBA (Visual Basic for Application) และมีการดึงข้อมูลเรื่องระยะทางจาก Google Map โดยนำวิธีการเมต้าฮิว ด้วยวิธี 3-OPT มาประยุกต์ใช้ในการเขียนโปรแกรม ซึ่งข้อมูลที่นำมาใช้ในการเขียนโปรแกรมมีดังนี้

1. รถบรรทุกสินค้า บริษัทกรณีศึกษามีรถขนส่งสินค้า 3 ประเภทด้วยกัน คือ รถกระบะ 4 ล้อ รถ 6 ล้อ และรถ 10 ล้อ (งานวิจัยครั้งนี้เลือกศึกษารถประเภทรถ 6 ล้อ เท่านั้น) โดยมีข้อจำกัด ดังนี้

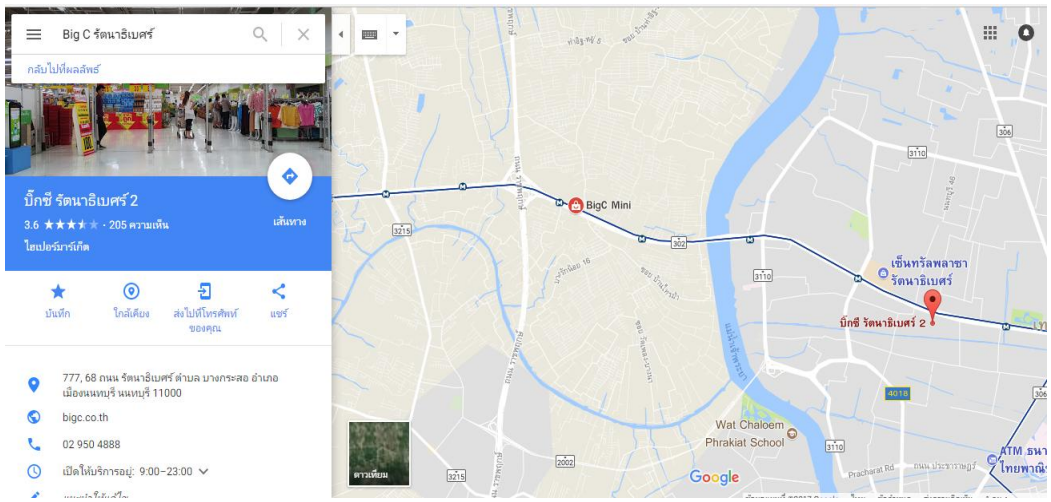
ตารางที่ 3.1 ประเภทและข้อจำกัดด้านน้ำหนักและปริมาตร

ประเภทรถ	ขนาดรถ (m)	น้ำหนักบรรทุกไม่รวมสินค้า	ปริมาตรบรรทุก
รถกระบะ 4 ล้อ	1.72*5.04*1.64	1,700	2.82
รถ 6 ล้อ	2.7 * 6.5 *2.5	7,500	10.8
รถ 10 ล้อ	2.8*10.3.1	15,600	34

2. ระยะทาง ข้อมูลระยะทางถือเป็นตัวข้อมูลจำเป็นที่ต้องประกอบในการจัดเส้นทางขนส่งสินค้า การได้มาซึ่งข้อมูลส่วนนี้นั้นมีหลากหลายวิธี ไม่ว่าจะเป็นมาจากข้อมูลจริง จากการคำนวณจากจุดละติจูด/ลองติจูด หรือได้จากการประยุกต์ใช้ระบบสารสนเทศทางภูมิศาสตร์ อย่างไรก็ตามการได้มาซึ่งระยะทางไม่ว่าจะวิธีใดล้วนแต่มีเป้าหมายในการจัดเส้นทางขนส่งสินค้าเดียวกันคือ ต้องการจัดเส้นทางเดินรถให้มีระยะทางที่ต่ำที่สุด

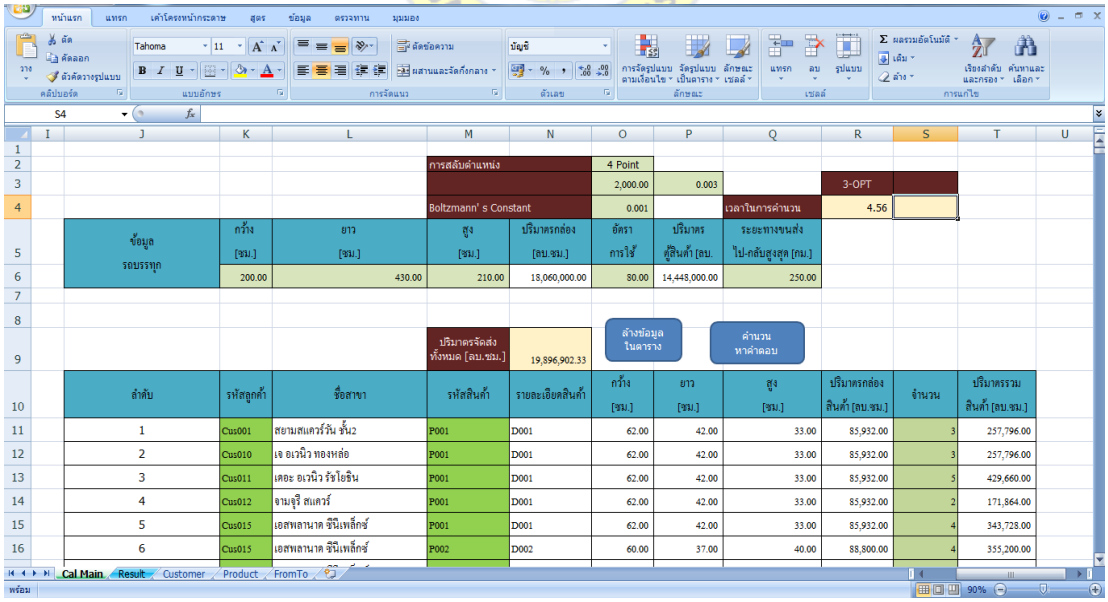
ในงานวิจัยนี้ ได้ใช้การคำนวณจากพิกัดละติจูดลองติจูด เป็นการคำนวณหาระยะทางระหว่างจุด 2 จุดโดยการนำพิกัดของจุด 2 จุดมาคำนวณหาระยะทางจากนั้นนำระบบสารสนเทศทางภูมิศาสตร์ (GIS) มาใช้ในการหาระยะทาง ในงานวิจัยนี้ได้เลือก Google Map ซึ่งเป็นโปรแกรมที่เกี่ยวกับระบบสารสนเทศทางภูมิศาสตร์อีกรูปแบบหนึ่งที่กำลังเป็นที่นิยมใช้ งานวิจัยครั้งนี้จึงต้องการที่จะวางแนวทางในการประยุกต์นำ (GIS) เข้ามาใช้ในการจัดเส้นทางเดินรถเพื่อเพิ่มประสิทธิภาพให้

มากยิ่งขึ้น ซึ่งจะจัดเก็บข้อมูลระยะทางจากศูนย์กระจายสินค้าไปยังลูกค้าแต่ละรายโดยมีศูนย์กระจายสินค้าเพียงแห่งเดียว



ภาพที่ 3.1 โปรแกรม Google Map

3. โปรแกรมที่ทำงานบน Microsoft Excal ร่วมกับ VBA (Visual Basic for Application) ถูกนำมาเขียนโค๊ดเพื่อจัดเส้นทางเดินรถขนส่งสินค้า



ภาพที่ 3.2 โปรแกรมที่ทำงานบน Microsoft Excel ร่วมกับ VBA (Visual Basic for Application

บทที่ 4
ผลการศึกษา

การวิจัยเรื่อง การจัดเส้นทางการขนส่งด้วยวิธีเมต้าฮิวริสติกส์ กรณีศึกษา บริษัทผู้ให้บริการขนส่งทางด้านโลจิสติกส์ ผู้วิจัยได้มีการปรับปรุงให้ใกล้เคียงและสอดคล้องกับปัญหาจริงให้มากขึ้นโดยมีการออกแบบจำลองดังนี้

4.1 การออกแบบการจัดเส้นทางเดินรถขนส่งสินค้า

งานวิจัยการจัดเส้นทางเดินรถขนส่งสินค้าได้มีการออกแบบจำลองการจัดเส้นทางเดินรถดังนี้

4.1.1 การสร้างเมตริกซ์ระยะทาง

งานวิจัยการจัดเส้นทางเดินรถขนส่งนี้มีการจัดเรียงข้อมูลระยะทางในรูปแบบของเมตริกซ์ซึ่งจากงานวิจัยต่าง ๆ ที่เกี่ยวกับการจัดเส้นทางเดินรถนั้นมีการเลือกใช้เมตริกซ์ระยะทาง 2 แบบด้วยกัน คือ เมตริกซ์แบบครึ่งชุด (Half - Matrix) และเมตริกซ์แบบเต็มชุด (Full - Matrix) โดยเมตริกซ์ทั้ง 2 แบบนั้นมีความแตกต่างกันที่แบบครึ่งชุดนั้นระยะทางจะมีเพียงครึ่งเมตริกซ์นั้นหมายถึง ระยะทางการวิ่ง ไป – กลับ นั้นมีระยะทางเท่ากัน รูปแบบเมตริกซ์แบบนี้จะง่ายต่อการคำนวณเนื่องจากสามารถลดจำนวนครั้งในการคิดคำนวณหาค่าประหยัดไปได้ถึงครั้งหนึ่ง ส่วนเมตริกซ์ระยะทางแบบเต็มชุดนั้นมีระยะทางในการวิ่ง ไป – กลับ ที่ไม่เท่ากันจึงมีการจัดเรียงข้อมูลเต็มเมตริกซ์ ในงานวิจัยนี้เลือกใช้การจัดเรียงข้อมูลระยะทางเมตริกซ์แบบเต็มชุดเพื่อให้สอดคล้องกับความเป็นจริงมากขึ้น เนื่องจากรถขนส่งในสถานการณ์จริงนั้นระยะทางการวิ่งรถ ไป – กลับ มีความเป็นไปได้ที่ระยะทางไม่เท่ากัน ซึ่งในเมตริกซ์ระยะทางนี้เป็นข้อมูลของระยะทางจากศูนย์กระจายสินค้าไปยังลูกค้ารายต่าง ๆ และระยะทางระหว่างลูกค้าในแต่ละราย หรือระยะทางจากจุด i ไป j โดยมีการกำหนดให้ระยะทางที่จุด i ไป j ที่เป็นตำแหน่งเดียวกันนั้นมีค่าเท่ากับ 0 ดังตารางเมตริกซ์ระยะทาง

ตารางที่ 4.1 ตัวอย่างเมตริกซ์ระยะทาง

	D001	Cus001	Cus002	Cus003	Cus004	Cus005	Cus006	Cus007	Cus008	Cus009	Cus010
D001	0.00	24.41	42.93	13.81	26.92	28.07	52.29	22.44	24.08	46.13	19.01
Cus001	27.70	0.00	20.09	30.29	20.58	2.40	26.87	29.73	18.24	38.12	9.64
Cus002	42.85	20.42	0.00	45.43	32.38	20.17	14.28	44.87	35.31	51.47	23.47
Cus003	12.74	25.21	44.58	0.00	29.09	31.52	53.08	11.62	13.58	35.36	21.92
Cus004	26.53	19.19	30.42	29.47	0.00	17.37	44.78	33.16	27.11	48.67	17.92
Cus005	27.74	2.61	19.51	31.87	18.83	0.00	24.83	31.31	20.02	36.36	7.89
Cus006	51.20	26.61	17.00	53.79	43.64	24.23	0.00	45.10	35.40	57.53	39.51
Cus007	24.67	22.84	40.59	12.66	35.99	25.73	46.13	0.00	8.21	20.57	24.24
Cus008	23.28	15.82	33.33	13.96	30.05	19.02	42.54	8.60	0.00	28.64	13.47
Cus009	44.25	32.06	49.82	25.04	48.57	34.96	55.06	18.02	23.52	0.00	34.35
Cus010	18.39	8.24	23.20	21.16	19.33	10.14	39.73	20.60	14.55	36.70	0.00

จากตารางที่ 4.1 ระยะทางที่บรรจุลงในตารางเมตริกซ์นี้ได้จากการประยุกต์ระบบสารสนเทศทางภูมิศาสตร์โดยการดึงข้อมูลระยะทางจาก Google Map มาจัดเก็บไว้ในโปรแกรมทางคอมพิวเตอร์

4.1.2 การประยุกต์ใช้วิธี 3-OPT ในการปรับปรุงเส้นทางเดินรถขนส่งสินค้า

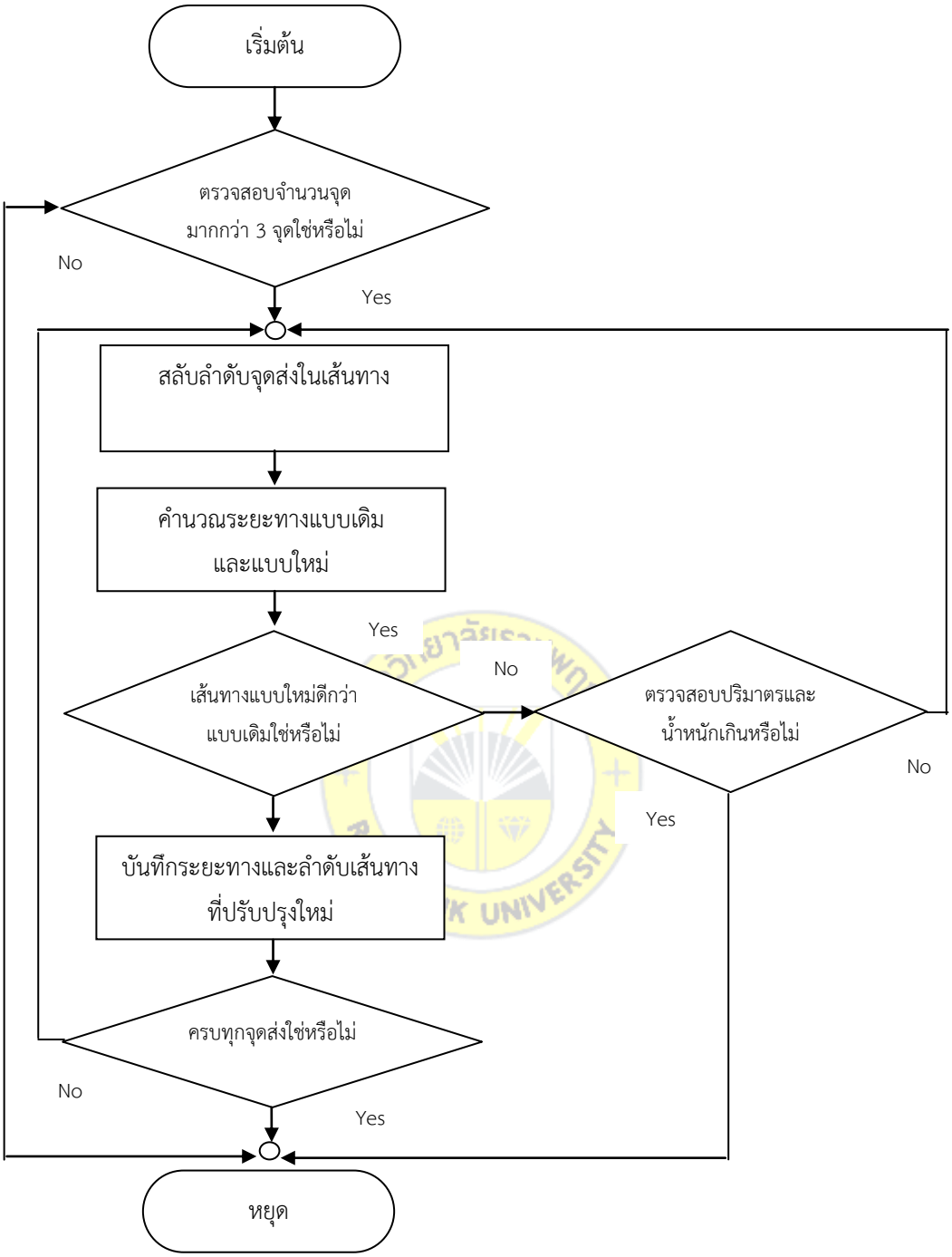
การปรับปรุงเส้นทางเดินรถขนส่งสินค้า คือการนำเส้นทางจากเดิมของบริษัทที่จัดโดยพนักงานขับรถมาทำการปรับเปลี่ยนปรับปรุงเพื่อให้ได้ผลลัพธ์ลดระยะทางในการขนส่งน้อยลงจากวิธีการเดิม การปรับปรุงเส้นทางนั้นถูกนำมาใช้ในการจัดเส้นทางเดินรถอย่างแพร่หลายโดยการประยุกต์เทคนิคฮิวริสติกส์ต่าง ๆ มาใช้ ในงานวิจัยนี้มีการปรับปรุงเส้นทางโดยใช้วิธี 3 – OPT หรือวิธีการสลับจุดส่งทีละ 3 จุด โดยวิธีการนี้จะเป็นวิธีการสลับลำดับจุดส่งเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดเมื่อนำมาเปรียบเทียบกับวิธีปัจจุบันของบริษัทที่ใช้อยู่ และนำข้อมูลที่ได้จากมาประยุกต์ใช้ในการหาคำตอบผ่านโปรแกรม Visual Basic for Application (VBA) ใน Microsoft Excel และดึงข้อมูลระยะทางด้วยระบบสารสนเทศทางภูมิศาสตร์ซึ่งมีขั้นตอนในการปรับปรุงดังนี้

ขั้นตอนที่ 1 สร้างเส้นทางเริ่มต้นก่อนโดยเลือกทุกจุดที่เป็นไปได้

ขั้นตอนที่ 2 ปรับปรุงเส้นทางโดยการสลับตำแหน่งจุดส่งใหม่โดยใช้วิธี 3 – OPT โดยลบเส้นทางที่มีอยู่ 3 เส้นทาง จากนั้นแทนที่เส้นทางใหม่แทนเส้นทางเดิม ที่ทำให้เส้นทางผ่านทุกจุดส่ง หลังจากนั้นตรวจสอบเส้นทางใหม่ที่ปรับปรุงว่ามีค่าใช้จ่ายในการขนส่งที่ต่ำกว่าเส้นทางเดิมหรือไม่ ถ้าเส้นทางที่ได้ใหม่มีค่าต่ำกว่าเส้นทางทางเดิม แสดงว่าเส้นทางใหม่ที่ได้เป็นคำตอบที่ดีกว่า

ขั้นตอนที่ 3 หลังจากทำการสลับลำดับจุดส่งเรียบร้อยแล้วและไม่ขัดแย้งกับเงื่อนไขหรือข้อกำหนดในการขนส่งแล้วนั้นจะทำการตรวจสอบระยะทางที่ได้จากการปรับปรุงเพื่อเปรียบเทียบกับระยะทางที่ได้ก่อนปรับปรุง หากผลที่ได้จากการปรับปรุงนั้นได้ค่าระยะทางที่ดีกว่าการจัดเส้นทางในขั้นตอนแรก เส้นทางที่ได้จากการปรับปรุงจะถูกบันทึกไว้

ขั้นตอนที่ 4 เมื่อได้เส้นทางที่เกิดจากการสลับลำดับจุดส่งและได้ค่าที่ประหยัดดีขึ้นแล้ว ทำการสลับลำดับจุดส่งเช่นนี้ไปเรื่อย ๆ เพื่อเปรียบเทียบกับเส้นทางที่ถูกบันทึกไว้หากเจอลำดับจุดส่งที่ให้ค่าประหยัดกว่าเส้นทางที่ถูกบันทึกไว้ก่อนหน้าให้บันทึกเส้นทางใหม่และทำซ้ำจนกระทั่งเกิดการสลับครบทุกจุด ดังภาพที่ 4.1



ภาพที่ 4.1 การจัดเส้นทางการเดินรถด้วยวิธี 3-OPT

4.2 แบบจำลองทางคณิตศาสตร์

งานวิจัยมีรูปแบบปัญหาการจัดเส้นทางเดินรถแบบ VRP จึงสามารถเขียนแบบจำลองทางคณิตศาสตร์ได้ดังต่อไปนี้

กำหนดให้ ค่าคงที่

N = จำนวนจุดส่ง

K = จำนวนรถบรรทุก

S_{ij} = ระยะทางการเดินทางจากจุดส่ง i ไปยังจุดส่ง j

W_j = น้ำหนักสินค้าที่ขนส่งไปยังจุดส่ง j

V_j = ปริมาตรสินค้าที่ขนส่งไปยังจุดส่ง j

W_k = น้ำหนักของสินค้าที่รถบรรทุก k สามารถบรรทุกได้

V_k = ปริมาตรของสินค้าที่รถบรรทุก k สามารถบรรทุกได้

ตัวแปรการตัดสินใจ

X_{ijk} = 1 เมื่อเดินทางจากจุดส่ง i ไปยังจุดส่ง j โดยรถบรรทุก k หากเป็นกรณีอื่นเท่ากับ 0

U_i = ตัวแปรสำหรับป้องกันการเกิดรอบการขนส่งย่อย (Sub tour) ที่อาจเกิดขึ้นในคำตอบ

หาค่าต่ำสุด

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum^{K} S_{ij} X_{ijk} \tag{4.1}$$

ภายใต้เงื่อนไข

$$\sum_{\substack{i=1\\ i\neq j}}^{N}\sum_{k=1}^{K} X_{ijk=1} \quad ; j = 2,.....,N \tag{4.2}$$

$$\sum_{\substack{i=1\\ i\neq j}}^{N}\sum_{k=1}^{K} X_{ijk=1} \quad ; i = 2,.....,N \tag{4.3}$$

$$\sum_{i=1}^{N} X_{ihk} - \sum_{j=1}^{N} X_{hjk} = 0 \qquad ; h = 1,.....,N, k = 1,.....k \tag{4.4}$$

$$\sum_{j=2}^{N}\sum_{\substack{i=1\\ i\neq j}}^{N} w_k X_{ijk} \le w_k \qquad ; k = 1,.....k \tag{4.5}$$

$$\sum_{j=2}^{N}\sum_{\substack{i=1\\ i\neq j}}^{N} v_k X_{ijk} \le v_k \qquad ; k = 1,.....k \tag{4.6}$$

$$(N-3) X_{jik} + (N-1) X_{jik} + U_i \le U_j + (N-2); i = 2,....,N; j = 2,...,N; k = 1,...,k \tag{4.7}$$

$$X_{ijk} \in (0,1); i = 1,\ldots,N; j = 1,\ldots,N; k = 1,\ldots,k \quad (4.8)$$

ฟังก์ชันวัตถุประสงค์

หาค่าระยะทางในการขนส่งที่น้อยที่สุด

ภายใต้เงื่อนไข

สมการที่ (4.2) กำหนดให้การวิ่งเข้าจุดส่งสินค้าในแต่ละจุดอนุญาตให้รถวิ่งเข้าเพียง 1 คัน เท่านั้น

สมการที่ (4.3) กำหนดให้การวิ่งออกจุดส่งสินค้าในแต่ละจุดอนุญาตให้รถออกเพียง 1 คัน เท่านั้น

สมการที่ (4.4) กำหนดให้แต่ละจุดส่งสินค้าเมื่อมีรถบรรทุกวิ่งเข้าแล้วต้องมีการวิ่งออกจากจุดส่งนั้น ๆ ด้วย

สมการที่ (4.5) กำหนดให้การบรรทุกสินค้าของรถบรรทุกแต่ละคันมีน้ำหนักรวมไม่เกินน้ำหนักที่รถบรรทุกสามารถรับได้

สมการที่ (4.6) กำหนดให้การบรรทุกสินค้าของรถบรรทุกแต่ละคันมีปริมาตรรวมไม่เกินปริมาตรที่รถบรรทุกสามารถรับได้

สมการที่ (4.7) เป็นการป้องกันไม่ให้เกิดเส้นทางเดินรถย่อย (Sub tour)

สมการที่ (4.8) กำหนดตัวแปรตัดสินใจการตัดสินใจ X_{ijk} มีค่าเป็น 0 หรือ 1 เท่านั้น

4.3 ต้นทุนน้ำมันเชื้อเพลิง

ในส่วนของต้นทุนค่าใช้จ่ายในการขนส่งของบริษัทกรณีศึกษา พบว่ารถบรรทุกสินค้ามีต้นทุนค่าขนส่งคือ ค่าก๊าซ LPG และค่าน้ำมันเชื้อเพลิง แต่ผู้วิจัยจะช่วยแก้ปัญหาการคิดลดต้นทุนค่าน้ำมันเชื้อเพลิงเท่านั้น ซึ่งต้นทุนน้ำมันต่อหน่วยผลิตนั้นสามารถคำนวณได้ดังต่อไปนี้

ต้นทุนน้ำมัน/หน่วยผลิต = (อัตราการใช้) x (ราคาน้ำมัน/หน่วย)

4.4 ผลการทดลองการจัดเส้นทางเดินรถขนส่งสินค้า

ผลการทดลองในงานวิจัยนี้ได้จากการทดลองจากกลุ่มตัวอย่างของบริษัทกรณีศึกษาโดยเลือกชุดข้อมูลมาทำการทดลองเป็นข้อมูลช่วงวันที่ 6 ถึง 12 พฤศจิกายน 2560 และเป็นจุดส่งที่อยู่ในเขตกรุงเทพมหานครและปริมณฑล ขนาดของข้อมูลมีจำนวนจุดส่งแตกต่างกันไป จากการนำชุดข้อมูลมาทดลองพบว่าผลลัพธ์ที่ได้จากการจัดเส้นทางเดินรถด้วยวิธีการจัดโดยใช้พนักงานจัดมาทำการเปรียบเทียบกับผลลัพธ์ที่ได้จากการจัดเส้นทางเดินรถโดยใช้โปรแกรมจัดเส้นทางที่เขียนชุดคำสั่งลง

บนโปรแกรม Visual Basic for Application (VBA) และทำงานบนโปรแกรม Microsoft Excel และดึงข้อมูลระยะทางด้วยระบบสารสนเทศทางภูมิศาสตร์ สามารถลดระยะทางขนส่งรวมจากเดิม 5,005.93 กิโลเมตร เหลือเพียง 4,512.11 กิโลเมตร โดยสามารถลดระยะทางจากเดิมลง 493.82 กิโลเมตร รายละเอียดแสดงดังตารางที่ 4.2 และสามารถลดต้นทุนค่าน้ำมันเชื่อเพลิงสามารถคำนวณได้ดังต่อไปนี้

ต้นทุนน้ำมันเชื้อเพลิง = (อัตราการใช้) x (ราคาน้ำมัน/หน่วย)

<u>วิธีเดิม</u>

ต้นทุนน้ำมันเชื้อเพลิงในการส่งสินค้า 7 วัน = 5,005.93 * 25.09

= 125,598.78 บาท

<u>วิธี 3-Opt</u>

ต้นทุนน้ำมันเชื้อเพลิงในการส่งสินค้า 7 วัน = 4,512.11 * 25.09

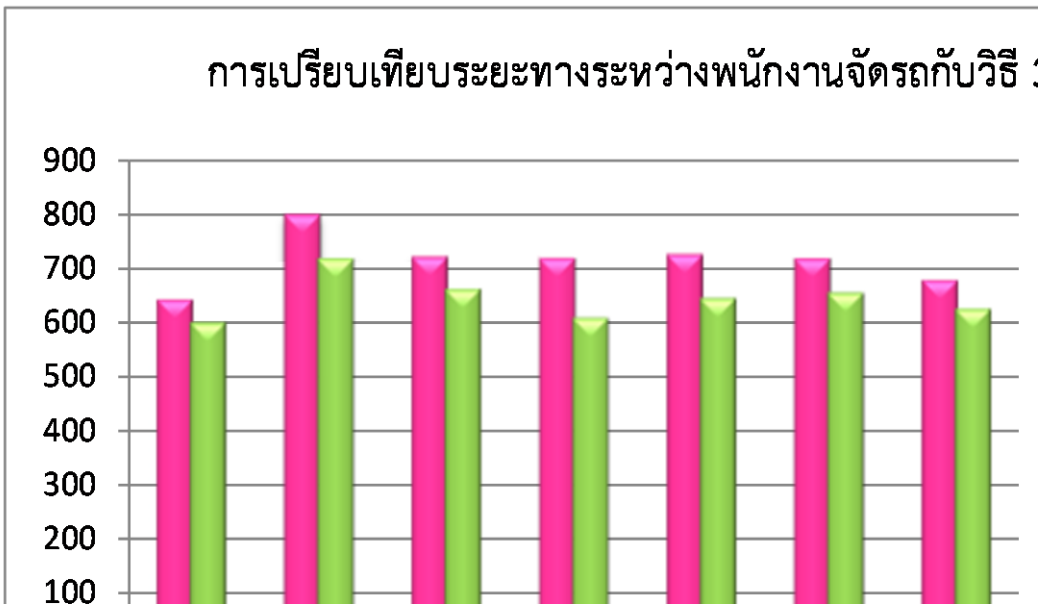
= 113,208.84 บาท

ดังนั้นในการขนส่งสินค้าจำนวน 7 วัน สามารถลดต้นทุนค่าน้ำมันเชื้อเพลิงลงได้

125, 598.78 - 113,208.84 = 12,389.94 บาท

ตารางที่ 4.2 เปรียบเทียบระยะทางหลังจากการปรับปรุงเส้นทาง

การเปรียบเทียบผล		**ชุดข้อมูลที่ 1 6 พ.ย.60**	**ชุดข้อมูลที่ 2 7 พ.ย.60**	**ชุดข้อมูลที่ 3 8 พ.ย.60**	**ชุดข้อมูลที่ 4 9 พ.ย.60**	**ชุดข้อมูลที่ 5 10 พ.ย. 60**	**ชุดข้อมูลที่ 6 11 พ.ย. 60**	**ชุดข้อมูลที่ 7 12 พ.ย.60**	**รวม ระยะทาง (KM)**
ระยะทาง (km)	พนักงานจัดรถ	642.91	799.86	721.66	719.19	725.78	718.41	678.12	5,005.93
	ปรับปรุงด้วยวิธี 3 – OPT	599.91	717.91	661.65	608.06	644.35	654.84	625.33	4,512.11
เปรียบเทียบระยะทาง		-42.94	-81.95	-60.01	-111.13	-81.43	-63.57	-52.79	-493.82



ภาพที่ 4.2 การเปรียบเทียบระยะทางระหว่างพนักงานจัดรถกับวิธี 3-OPT

มหาวิทยาลัยราชพฤกษ์ RAJAPRUK UNIVERSITY

บทที่ 5
สรุป อภิปรายผล และข้อเสนอแนะ

การวิจัยครั้งนี้มีจุดประสงค์เพื่อนำเทคนิคในการจัดเส้นทางการขนส่งด้วยวิธีการเมต้าฮิวริสติกส์มาใช้ในการจัดเส้นทางการขนส่งสินค้าให้กับบริษัทกรณีศึกษา และช่วยในการลดระยะทางในการขนส่งให้ได้ระยะทางที่สั้นที่สุดเป็นงานวิจัยที่จัดทำขึ้นเพื่อประยุกต์หลักการจัดเส้นทางเดินรถขนส่งสินค้าด้วยวิธีเมต้าฮิวริสติกส์มาใช้กับปัญหาจริงและสร้างโปรแกรมจัดรถขึ้นเพื่อเป็นแนวทางให้กับบริษัทกรณีศึกษาในการพัฒนาเทคโนโลยีเข้ามาใช้เป็นเครื่องมือช่วยในการตัดสินใจให้กับพนักงานจัดรถเพื่อให้การทำงานเกิดประสิทธิภาพเพิ่มมากขึ้น

5.1 สรุปผลการวิจัย

การวิจัยครั้งนี้ได้มีการออกแบบจำลองการจัดเส้นทางเดินรถขนส่งสินค้าโดยได้นำหลักการเมต้าฮิวริสติกส์ด้วยวิธีการสลับจุด 3 จุด 3-OPT มาประยุกต์กับโปรแกรมจัดเส้นทางที่เขียนชุดคำสั่งลงบนโปรแกรม Visual Basic for Application (VBA) และทำงานบนโปรแกรม Microsoft Excel และดึงข้อมูลระยะทางด้วยระบบสารสนเทศทางภูมิศาสตร์ โดยการเก็บข้อมูลตัวอย่าง 7 ชุดข้อมูลระหว่างวันที่ 6 -12 พฤศจิกายน 2560 ทดลองจัดเส้นทางด้วยวิธี 3-OPT โดยเลือกรถบรรทุกประเภท 6 ล้อกลาง ผลที่ได้จากการทดลองสามารถลดระยะทางขนส่งรวมจากเดิม 5,005.93 กิโลเมตร เหลือเพียง 4,512.11 กิโลเมตร โดยสามารถลดระยะทางจากเดิมลง 493.82 กิโลเมตร และยังสามารถลดต้นทุนค่าน้ำมันเชื้อเพลิงลงได้ 7 วัน 12,389.94 บาทรายละเอียดดังตารางที่ 5.1

ตารางที่ 5.1 เปรียบเทียบระยะทางหลังจากการปรับปรุงเส้นทาง

การเปรียบเทียบผล		ชุดข้อมูลที่ 1 6 พ.ย.60	ชุดข้อมูลที่ 2 7 พ.ย.60	ชุดข้อมูลที่ 3 8 พ.ย.60	ชุดข้อมูลที่ 4 9 พ.ย.60	ชุดข้อมูลที่ 5 10 พ.ย.60	ชุดข้อมูลที่ 6 11 พ.ย.60	ชุดข้อมูลที่ 7 12 พ.ย.60	รวมระยะทาง (KM)
ระยะทาง (km)	พนักงานจัดรถ	642.91	799.86	721.66	719.19	725.78	718.41	678.12	5,005.93
	ปรับปรุงด้วยวิธี 3 – OPT	599.91	717.91	661.65	608.06	644.35	654.84	625.33	4,512.11
เปรียบเทียบระยะทาง		-42.94	-81.95	-60.01	-111.13	-81.43	-63.57	-52.79	-493.82

5.2 อภิปรายผลการวิจัย

ในการวิจัยเรื่องนี้มีข้อค้นพบที่เด่น ๆ เห็นควรนำมาอภิปรายโดยใช้หลักตรรกวิทยาและอภิปรายผลการวิจัยให้ครอบคลุมวัตถุประสงค์ของการวิจัยทุกข้อทุกประเด็น ดังต่อไปนี้

ผลการวิจัยพบว่า เมื่อนำทฤษฎีเทคนิคทางเมต้าฮิวริสติกส์มาประยุกต์ใช้ในการออกแบบจำลองการจัดเส้นทางเดินรถขนส่งสินค้ามาประยุกต์กับโปรแกรมจัดเส้นทางที่เขียนชุดคำสั่งลงบนโปรแกรม Visual Basic for Application (VBA) และทำงานบนโปรแกรม Microsoft Excel ก็สามารถลดระยะทางขนส่งลงได้เมื่อนำมาเปรียบเทียบกับวิธีการจัดเส้นทางแบบเดิมของบริษัทซึ่งสอดคล้องกับงานวิจัยของ วิไลวรรณ์ แก่นสาร และ สมบัติ สินธุเชาน์ (2556) ได้ศึกษาการเปรียบเทียบวิธีการฮิวริสติกส์ สำหรับระบบการจัดการขยะซึ่งจัดทำเพื่อแก้ปัญหาในการลดต้นทุนต่าง ๆ ทั้งทางตรงและทางอ้อม พร้อมทั้งเกิดการลดต้นทุนค่าใช้จ่ายได้เป็นอย่างดีอีกทั้งยังมีวิธีการที่เข้าใจง่ายไม่ซับซ้อนโดยเนื้อหาใน งานวิจัยนี้ได้บอกถึงการประยุกต์วิธีการหาคำตอบโดยวิธีการ Saving heuristic , Nearest heuristic และ Max-Nearest เพื่อทำการเปรียบเทียบกันโดยพิจารณาจากผลลัพธ์ที่ได้ภายใต้เงื่อนไขเดียวกัน ผลทดลองของงานวิจัยนี้จะเห็นว่าวิธี Saving heuristic ให้คำตอบที่ดีที่สุดคือมีระยะทางรวม 55.022 กิโลเมตร และวิธี Nearest heuristic ต่อด้วยวิธี Max-nearest คือ 62.152 กิโลเมตร และ 67.196 กิโลเมตร ตามลำดับ ซึ่งในปัจจุบันเส้นทางในการเก็บขยะจะมีระยะทางรวมอยู่ที่ 62.246 กิโลเมตร ดังนั้นจาก การทดลองจะเห็นว่าวิธี Saving และ Nearest สามารถลดระยะทางลงได้พร้อมกับสามารถควบคุม เวลาในการเก็บขยะได้อีกด้วย และสอดคล้องกับงานวิจัยของ ปาลีรัฐ บุญก่อน (2554) ได้ศึกษาการพัฒนาระบบการจัดเส้นทางขนส่งสินค้าจากศูนย์กระจายสินค้าไปยังลูกค้าตามจุดต่างๆ เพื่อช่วยสนับสนุนการตัดสินใจผู้ใช้ระบบ โดยมีศูนย์กระจายสินค้าแห่งเดียวและมีรูปแบบปัญหาการจัดเส้นทางการเดินรถแบบมีข้อจำกัดเรื่องความสามารถในการบรรทุกโดยพิจารณาเรื่องน้ำหนักและปริมาตรสินค้าให้อยู่ภายใต้เงื่อนไขที่จำกัดเรื่องความสามารถในการบรรทุกโดยพิจารณาเรื่องน้ำหนักและปริมาตรสินค้าให้อยู่ภายใต้เงื่อนไขของความจุรถโดยมีเป้าหมายเพื่อให้ได้ระยะทางที่สั้นที่สุด วิธีที่ใช้ในการจัดเส้นทางเดินรถขนส่งสินค้าของงานวิจัยนี้ คือ วิธีฮิวริสติกส์แบบ Saving Algorithm ของ Clarke and Wright โดยใช้วิธี 2 –OPT ปรับปรุงเส้นทางและมีการประยุกต์ใช้ระบบสารสนเทศทางภูมิศาสตร์ร่วมด้วยโดยจัดทำโปรแกรมลงบน Visual Basic for Application (VBA) ใน Microsoft Excel ผลลัพธ์ที่แสดงออกมาสามารถระบุตำแหน่งจุดส่งสินค้าปริมาตรและน้ำหนักสินค้าที่รถบรรทุกแต่ละคันบรรทุกไปซึ่งผลลัพธ์ที่ได้สามารถเพิ่มประสิทธิภาพอรรถประโยชน์การใช้รถมากขึ้นภายในระยะเวลาสั้นลงและสามารถลดเวลาที่ใช้ในการจัดรถได้ดีเมื่อเปรียบเทียบกับข้อมูลตัวอย่าง

5.3 ข้อเสนอแนะ

ข้อค้นพบจากการวิจัยและการอภิปรายผลมีข้อเสนอแนะที่ควรนำไปใช้ดังนี้

5.3.1 ข้อเสนอแนะในการนำผลการวิจัยไปใช้

5.3.1.1 งานวิจัยที่จัดทำขึ้นเพื่อเป็นแนวทางในการพัฒนาการจัดเส้นทางเดินรถให้กับบริษัทกรณีศึกษาซึ่งสามารถนำไปใช้ได้แต่ต้องมีการประยุกต์ให้กับการระบบการทำงานของบริษัทก่อนจึงควรมีการจัดทำฐานข้อมูลเพิ่มเติม

5.3.2 ข้อเสนอแนะในการทำวิจัยครั้งต่อไป

5.3.2.1 ควรมีการเก็บข้อมูลเพิ่มในเรื่องของรายละเอียดของสินค้า ปริมาตรของตัวสินค้า น้ำหนักของตัวสินค้า ที่อยู่ของลูกค้า ตำแหน่งพิกัดละติจูด/ลองติจูด

5.3.2.2 ควรมีการวิเคราะห์ให้ครอบคลุมถึงเรื่องของต้นทุนมีการเพิ่มตัวแปรในการขนส่งโดยคิดเป็นบาทต่อกิโลเมตรและครอบคลุมถึงเรื่องต้นทุนทางด้านเวลา (Time cost) ด้วย

มหาวิทยาลัยราชพฤกษ์ RAJAPRUK UNIVERSITY

บรรณานุกรม

ปฏิพัทธ์ หงส์สุวรรณ วิชัย จันทรรักษา และ สรวิทย์ เชื้อพิสุทธ์กุล (2556). **“การศึกษาเพื่อเพิ่มประสิทธิภาพการจัดเส้นทางเดินรถขนส่งน้ำดื่ม จังหวัดสมุทรสงคราม”.**ภาควิชาวิศวกรรมอุตสาหการและการ จัดการ คณะวิศกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปกรนครปฐม.

พัฒนพงษ์ สุหญ้านาง (2552). **“การจัดเส้นทางการขนส่งที่เหมาะสมในระบบมิลค์รัน**”. วิศวกรรมการจัดการอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.

ทวินันท์ สิมะจารึก และคณะ (2552). **“การลดค่าใช้จ่ายในการขนส่ง กรณีศึกษา โรงงานเคมีภัณฑ์”.** ภาควิชาวิศวกรรมอุตสาหการและการจัดการ คณะวิศกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยศิลปกรนครปฐม

อภิชิต มณีงาม (2556). **“การแก้ปัญหาการจัดเส้นทางการเดินรถโดยมีการจำกัดเวลาการเดินรถบรรทุกขนาดใหญ่ด้วยวิธีฮิวริสติกส์ กรณีศึกษา การขนส่งอิฐบล็อกในเขตกรุงเทพมหานครและปริมณฑล”.**คณะวิศกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ไพโรจน์ แสนดี และคณะ. 2557. **“การศึกษาเส้นทางเดินรถในการเคลื่อนย้ายผู้ประสบอุทกภัยออกจากพื้นที่อันตรายเมื่อระดับน้ำสูง”**วารสารวิชาการอุตสาหกรรมศึกษา. 8, 1 (มกราคม-มิถุนายน 2557)

วัลลภ รัตนถาวร และ สีรง ปรีชานนท์. 2553. **“ปัญหาตำแหน่งที่ตั้งศูนย์กระจายสินค้าแบบพลวัตโดยมีข้อจำกัดด้านขนาด”** วารสารวิศวกรรมศาสตร์. 2,1 (กุมพาพันธ์2553)

วิไลวรรณ์ แก่นสาร และ สมบัติ สินธุเชาน์. 2556. **“การเปรียบเทียบวิธีการฮิวริสติกส์สำหรับระบบการจัดการขยะ”** วารสารวิชาการ Thai VCML. 6, 2 (ธันวาคม 2556).

อตินุช ลิมป์มณีรักษ์ และ นระเกณฑ์ พุ่มชูศรี. 2556. **“การออกแบบระบบสารสนเทศสำหรับองค์กร กลาง ให้บริการการขนส่ง”** วารสารวิศวกรรมศาสตร์ ISSN: 1906-3636. 5,1 (มกราคม 2557)

อภิชิต มณีงาม และคณะ.2556. **“การแก้ปัญหาการจัดเส้นทางการเดินรถโดยมีการจำกัดเวลาการเดินรถบรรทุกขนาดใหญ่ด้วยวิธีฮิวริสติกส์”** วารสารเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏอุบลราชธานี. 3, 6 (กรกฎาคม – ธันวาคม 2556)

อรประไพ จารุพัฒน์ และ ปวีณา เชาวลิตวงศ์. 2556. **“ฮิวริสติกสำหรับการจัดเส้นทางเดินรถแบบ เปิดเพื่อ ลดต้นทุนการขนส่ง”**วารสารวิศวกรรมศาสตร์. (ISSN: 1906-3636) 4,3 (31 พฤษภาคม 2556)

พลอยพรรณ ศรีกิจการ และ อรอุไร แสงสว่าง. 2556. **“การออกแบบเพื่อเพิ่มประสิทธิภาพเส้นทางเดินรถขนส่งเครื่องสำอาง”** วารสารวิชาการอุตสาหกรรมศึกษา. 7, 2 (ธันวาคม 2556)

ธรินี มณีศรี.2550 **“การพัฒนาอัลกอริทึมสำหรับปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีหลายคลังสินค้า และมีความไม่แน่นอนภายใต้กรอบเวลา”**.สาขาวิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีปทุม.

จตุรวิทย์ ศศิธรานนท์และธราธร กูลภัทรนิรันดร์.2550. **“การประยุกต์วิธีเชิงพันธุกรรมสำหรับการแก้ปัญหาการจัดการการขนส่งสินค้า กรณี มีข้อจำกัดด้านเวลา”.**สาขาเทคโนโลยีโลจิสติกส์ มหาวิทยาลัยเทคโนโลยีมหานคร กรุงเทพ.

ชัยยา นุรักษ์เข (2541). **“การจัดเส้นทางเดินรถขนส่งขนมในเขตกรุงเทพฯและปริมณฑล กรณีศึกษา : ห้าง โอ.ซี.ซัพพลาย”.** สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ กรุงเทพ.

เครือวัลย์ จำปาเงิน (2547.) **“การพัฒนาระบบสนับสนุนการตัดสินใจเพื่อจัดเส้นทางเดินรถสำหรับการขนส่งสินค้าเพื่อการบริโภคสู่ร้านค้าปลีกในสถานบริการน้ำมันในจังหวัดนนทบุรี”.** สาขาอักษรศาสตรมหาบัณฑิต จุฬาลงกรณ์มหาวิทยาลัย.

สุดารัตน์ สุ่มมาตย์ (2548). **“การใช้วิธีเชิงฮิวริสติกส์เพื่อแก้ปัญหาการจัดเส้นทางยานพาหนะและการบรรจุ”.**สาขาวิศวกรรมศาสตร์ (วิศวกรรมอุตสาหการ) มหาวิทยาลัยเกษตรศาสตร์.

อุดม จาปะเกษตร์ (2542). **“การศึกษาการจำลองสถานการณ์ การจัดเส้นทางยานพาหนะสำหรับศูนย์กระจายสินค้าอะไหล่รถยนต์”.**บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์.

ชัยวัฒน์ สุขไมตรี (2550). **“การเพิ่มประสิทธิภาพการจัดเส้นทางการขนส่งสินค้าโดยวิธีมูลค่า”** สารนิพนธ์วิทยาศาสตรมหาบัณฑิต สาขาการจัดการการขนส่งของโลจิสติกส์ มหาวิทยาลัยบูรพา

ผกาวดี แสงสุวรรณ (2555). **“ขั้นตอนวิธีเชิงพันธุกรรมแบบหลายวัตถุประสงค์ที่มีวิวัฒนาการทำงานร่วมกันสำหรับแก้ปัญหาการจัดตารางสอน”** วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต (สถิติประยุกต์และเทคโนโลยีสารสนเทศ) คณะสถิติประยุกต์ สถาบันบัณฑิตพัฒนบริหารศาสตร์

อรประไพ จารุพัฒน์ และ ปวีณา เชาวลิตวงศ์. 2556. **“ฮิวริสติกสำหรับการจัดเส้นทางเดินรถแบบ เปิดเพื่อ ลดต้นทุนการขนส่ง”** วารสารวิศวกรรมศาสตร์. (ISSN: 1906-3636) 4,3 (31 พฤษภาคม 2556)

พิมพ์ชนก สุวรรณศรี (2558). **“การศึกษารูปแบบการจัดเส้นทางการขนส่งสินค้าด้วยวิธีเชิงพันธุกรรม”** โครงการวิจัยนี้ได้รับทุนสนับสนุนจาก มหาวิทยาลัยราชภัฏเชียงใหม่ งบประมาณ

แผ่นดิน ปีงบประมาณ พ.ศ. 2558 โดยการสนับสนุนทางวิชาการจากสานักงานคณะกรรมการวิจัยแห่งชาติ (วช.) 2558

ผศ.ดร.ระพีพันธ์ ปิตาคะโส.วิธีการเมต้าฮิวริสติกส์.พิมพ์ครั้งที่ 1.กรุงเทพมหานคร : สำนักพิมพ์ส.ส.ท.;2554.

ปาลีรัฐ.2554 **“แนวทางการพัฒนาการจัดเส้นทางเดินรถขนส่งสินค้า กรณีศึกษา ศูนย์กระจายสินค้าประเภท เซรามิค”**.สาขาวิชาการจัดการโลจิสติกส์ บัณฑิตวิทยาลัยการจัดการนวัตกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี.

กวี ศรีเมือง.2551. **“การหาจำนวนรถบรรทุกที่เหมาะสมในการขนส่งสินค้าในธุรกิจค้าปลีก กรณีศึกษาท็อปส์ซูปเปอร์มาร์เก็ต”** วิทยานิพนธ์วิศวกรรมศาสตรมหาบัรฑิต สาขาวิศวกรรมโยธาคณะวิศกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

วิไลวรรณ์ แก่นสาร และ สมบัติ สินธุเชาน์. 2556. **“การเปรียบเทียบวิธีการฮิวริสติกส์สำหรับระบบการจัดการขยะ”** วารสารวิชาการ Thai VCML. 6,2 (ธันวาคม 2556).

สาวนลินี อุดมสมบัติมีชัย. 2548. **“การประยุกต์วิธีศึกษาสานึกสาหรับการจัดเส้นทางยานพาหนะแบบมีกรอบเวลา”**. วิทยานิพนธ์บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์.

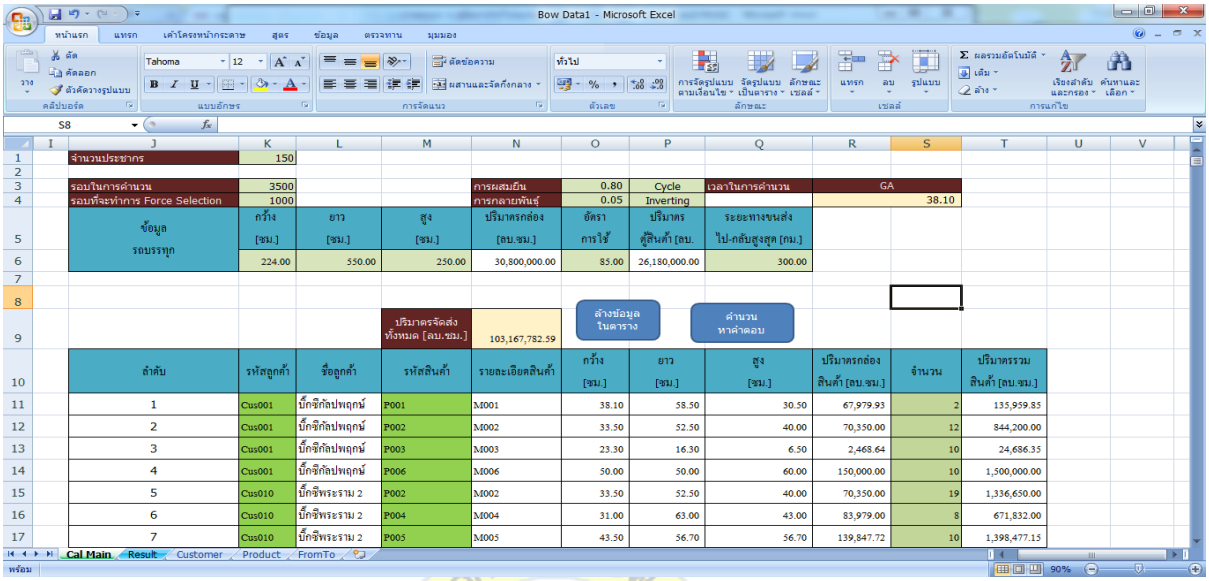
อนันต์ มุ่งวัฒนา และ ธรินี มณีศรี. 2555. **“การเปรียบเทียบขั้นตอนวิธีเมต้าฮิวริสติกสำหรับแก้ปัญหาการจัดเส้นทางเดินรถขนส่งแบบมีกรอบเวลากรณีมีรถขนส่งหลายขนาดและแบ่งแยกส่งสินค้าได้”**การประชุมวิชาการข่ายงานวิศวกรรมอุตสาหการ. (ตุลาคม 2555)

ภาคผนวก ก
คู่มือการใช้โปรแกรมการจัดเส้นทางเดินรถ

มหาวิทยาลัยราชภัฏ RAJAPRUK UNIVERSITY

ก.1 หน้าต่างโปรแกรมจัดเส้นทางเดินรถขนส่ง

โปรแกรมจัดเส้นทางเดินรถถูกเขียนโดย Visual Basic for Application: VBA มีการปฏิบัติการโปรแกรมบน Microsoft Excel ซึ่งมีหน้าต่างการทำงานดังรูปที่ ก.1



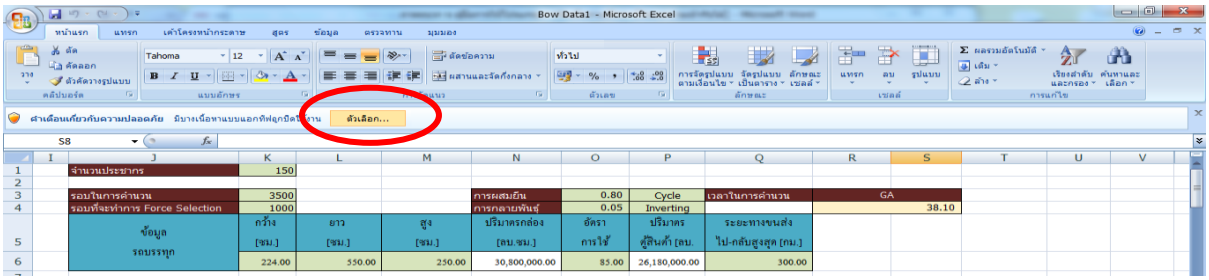
ภาพที่ ก.1 ตัวอย่างหน้าโปรแกรมจัดเส้นทางเดินรถ

ก.2 ขั้นตอนการใช้โปรแกรมจัดเส้นทางเดินรถขนส่ง

เนื่องจากโปรแกรมจัดเส้นทางเดินรถขนส่งจัดทำบน Microsoft Excel การจัดเก็บข้อมูลก็ถูกจัดเก็บไว้ใน Microsoft Excel เช่นเดียวกัน ส่วนตัวโปรแกรมเองจึงมีหลายหน้าต่างสำหรับการจัดเก็บข้อมูล การปรับค่าการแสดงค่าและการแสดงผลจากการจัดเส้นทางต่างๆ ดังนั้นในหัวข้อนี้จึงมีการอธิบายควบคู่ไปกับการแนะนำขั้นตอนการใช้งานโปแกรมเพื่อให้เข้าใจถึงหน้าที่ของแต่ละหน้าต่างการทำงาน (Worksheet)

1. ขั้นตอนการเปิดโปรแกรม

เมื่อเปิดโปรแกรมจะพบแถบ Security Warning ให้คลิกปุ่ม ตัวเลือก จากนั้นให้คลิก ⊙ เปิดใช้งานเนื้อหานี้ ไฟล์งานจึงจะสามารถแก้ไขเปลี่ยนแปลงข้อมูลได้ตามภาพที่ ก2



ภาพที่ ก.2 วิธีการใช้โปรแกรมจัดเส้นทางเดินรถ

2. อธิบายและแนะนำการใช้งานแต่ละหน้าต่างการทำงาน

2.1 การจัดเก็บข้อมูลลูกค้า (Worksheet: Customer)

ในการจัดเก็บข้อมูลลูกค้าในหน้าต่างชื่อว่า Customer รูปแบบการจัดเก็บข้อมูลจะประกอบไปด้วยชื่อลูกค้าข้อมูลเกี่ยวกับชื่อลูกค้าที่ปรากฏบนบิลรายการสินค้าโดยข้อมูลนี้จำเป็นต้องกรอกไว้ในฐานข้อมูล แต่ในงานวิจัยนี้จะทำการสมมติชื่อลูกค้าเพื่อใช้ในการทดสอบโปรแกรมพิกัด N/E ข้อมูลนี้เป็นข้อมูลพิกัดละติจูด/ลองติจูดของลูกค้าซึ่งข้อมูลนี้สำคัญมากต้องกรอกไว้ในฐานข้อมูล Worksheet: Customer ผู้ใช้ระบบจะต้องทำการกรอกข้อมูลลูกค้าซึ่งประกอบด้วย รหัสลูกค้า ชื่อลูกค้า ละติจูด ลองติจูดที่ตั้งลูกค้า ดังแสดงในภาพที่ ก.3

รหัสลูกค้า	ชื่อลูกค้า	ละติจูด	ลองติจูด
D001	PDep1	13.766785	100.528416
Cus1	PCus1	13.765579	100.527046
Cus2	PCus2	13.748650	100.583191
Cus3	PCus3	13.727898	100.531514
Cus4	PCus4	13.608731	100.551156
Cus5	PCus5	13.780511	100.616183
Cus6	PCus6	13.714339	100.489846
Cus7	PCus7	13.676058	100.501051
Cus8	PCus8	13.708792	100.545125
Cus9	PCus9	13.746409	100.509442
Cus10	PCus10	13.714050	100.587709
Cus11	PCus11	13.676697	100.606474
Cus12	PCus12	13.787036	100.509627

Cal Main | Result | Customer | Product | FromTo

ภาพที่ ก.3 โปรแกรมหน้า Worksheet: Customer

2.2 การจัดเก็บข้อมูลสินค้า (Worksheet: Product)

Worksheet: Product ผู้ใช้ระบบจะต้องทำการกรอกข้อมูลสินค้าซึ่งประกอบไปด้วยรหัสสินค้าเป็นช่องสำหรับการกรอกรหัสของสินค้าแต่ละตัวที่ต้องจัดเก็บเป็นข้อมูลสำคัญที่ต้องกรอกเก็บไว้ในฐานข้อมูลรายละเอียดสินค้าเป็นข้อมูลชื่อของสินค้าและข้อมูลอธิบายเกี่ยวกับตัวสินค้าของแต่ละตัวข้อมูลในส่วนนี้ควรจะจัดเก็บไว้ในฐานข้อมูลเพื่อบอกชื่อของสินค้าแต่ละตัวกว้าง ซม.เป็นข้อมูลเกี่ยวกับขนาดความกว้างของสินค้าแต่ละในหน่วยเซนติเมตร ข้อมูลส่วนนี้จำเป็นจะต้องจัดเก็บในฐานข้อมูลยาว ซม.เป็นข้อมูลเกี่ยวกับขนาดความยาวของสินค้าแต่ละในหน่วยเซนติเมตร ข้อมูลส่วนนี้จำเป็นจะต้องจัดเก็บในฐานข้อมูลสูง ซม.เป็นข้อมูลเกี่ยวกับขนาดความสูงของสินค้าแต่ละในหน่วยเซนติเมตรข้อมูลส่วนนี้จำเป็นจะต้องจัดเก็บในฐานข้อมูล

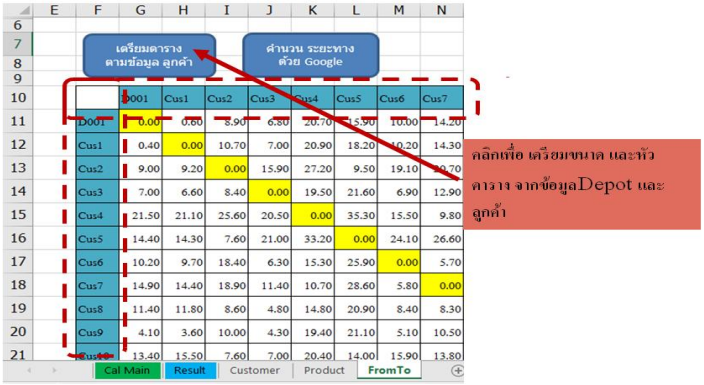
	E	F	G	H	I	J	K
9							
10		รหัสสินค้า	รายละเอียดสินค้า	กว้าง [ซม.]	ยาว [ซม.]	สูง [ซม.]	ปริมาตรกล่อง [ลบ.ซม.]
11		P001	M001	25.00	40.00	8.00	8,000.00
12		P002	M002	20.00	32.00	8.00	5,120.00
13		P003	M003	25.00	32.00	12.00	9,600.00
14		P004	M004	17.00	32.00	15.00	8,160.00
15		P005	M005	20.00	35.00	8.00	5,600.00
16		P006	M006	25.00	32.00	12.50	10,000.00
17		P007	M007	25.00	25.00	8.00	5,000.00
18		P008	M008	25.00	32.00	12.50	10,000.00
19		P009	M009	25.00	35.00	12.00	10,500.00
20		P010	M010	25.00	25.00	8.00	5,000.00
21		P011	M011	20.00	32.00	8.00	5,120.00
22		P012	M012	25.00	32.00	8.00	6,400.00
23		P013	M013	17.00	32.00	12.00	6,528.00

Cal Main | Result | Customer | Product | FromTo

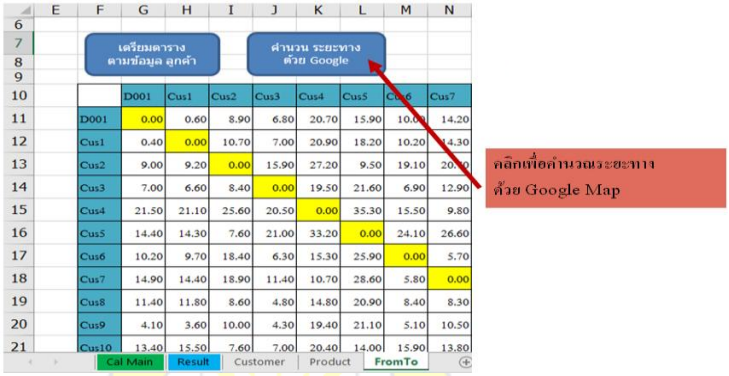
ภาพที่ ก.4 โปรแกรมหน้า **Worksheet: Product**

2.3 การดึงระยะทางจาก Google Map (Worksheet: From To)

หลังจากที่ผู้ใช้งานระบบกรอกข้อมูล Worksheet: Customer และ Worksheet: Product เรียบร้อยแล้วถ้าต้องการ คำนวณระยะทางด้วย Google Map ให้ไปที่หน้า Worksheet: From To แล้วกดคำนวณระยะทางด้วย Google Map หน้า Worksheet: From To จะแสดงการคำนวณระยะทางของลูกค้าโดยดึงข้อมูลจาก Google Map ซึ่งก่อนคำนวณระยะทางจะต้องกดเตรียมตารางข้อมูลลูกค้าก่อน หลังจากกดแล้วจะแสดงข้อมูลขนดลูกค้าตามที่เราได้กรอกไว้ในหน้า Worksheet: Customer ดังแสดงในภาพที่ ก.5 และ ภาพที่ ก.6



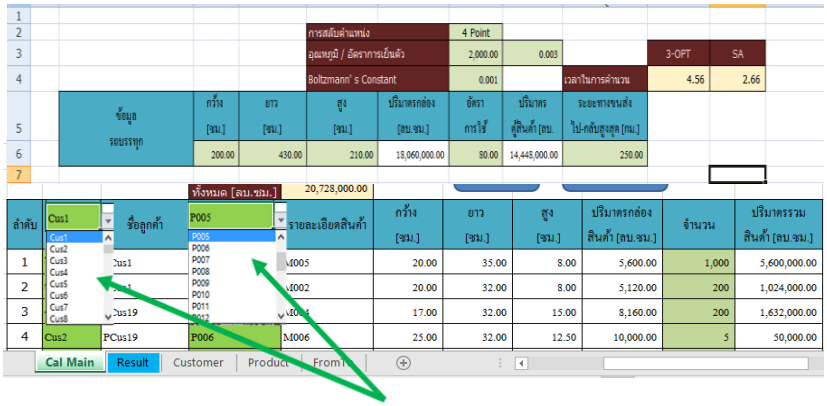
ภาพที่ ก.5 เตรียมตารางข้อมูลลูกค้า



ภาพที่ ก.6 คำนวณระยะทางด้วย Google Map

2.4 การคำนวณอัลกอริทึมและการป้อนข้อมูล

การคำนวณอัลกอริทึมจะกำหนดให้โปรแกรมมีการสลับตำแหน่ง 4 Point อุณหภูมิสูงสุดที่ 2,000และอัตราการเย็นตัวที่ 0.003 จากนั้นป้อนข้อมูลที่จะทำการจัดส่ง เช่น ข้อมูลลูกค้า หน้า Worksheet: Call Main จะทำการดึงข้อมูลจากหน้า Worksheet: Customer และ Worksheet: Product โดยสามารถคลิกปุ่ม Combo Box เลือกได้ ดังแสดงในภาพที่ ก.7



ภาพที่ ก.7 การคำนวณอัลกอริทึมและหน้า Worksheet: Call Main

2.4 การจัดเส้นทางเดินรถ (Worksheet: Result)

เมื่อกรอกข้อมูลการจัดส่งเรียบร้อยแล้วสามารถกดคำนวณหาคำตอบ โดยเมื่อกดคำนวณหาคำตอบแล้ว ผลลัพธ์ที่ได้จะไปแสดงข้อมูลในหน้า Worksheet: Result โดยในหน้านี้จะแสดงผลการคำนวณ เปรียบเทียบ วิธีการจัดส่งแบบเดิมของบริษัท กับวิธี 3-Opt โดยจะบอกรายละเอียดต่างๆไว้ประกอบไปด้วย

จำนวนรถ บอกว่าจำนวนรถทั้งหมดที่ใช้ในการขนส่ง

ระยะทาง บอกระยะทางในการขนส่งสินค้าโดยจะบอกเป็นระยะทางรวมในเส้นนั้น

น้ำหนัก บอกน้ำหนักที่รถแต่ละคันบรรทุกไป

ปริมาตร บอกปริมาตรที่รถแต่ละคันบรรทุกไป

					ระยะทางรวม	661.89
การขนส่งแบบเดิมของบริษัท						
รหัสลูกค้า	รหัสสินค้า	ปริมาตรกล่องสินค้า [ลบ.	จำนวน	ปริมาตรรวมสินค้า [ลบ.ซม.]	ขนส่งด้วยรถ	ระยะทาง
Cus042	P001	85,932	4	343,728	1	

					ระยะทางรวม	639.20
จัดแบบ 3 Opt						
รหัสลูกค้า	รหัสสินค้า	ปริมาตรกล่องสินค้า [ลบ.	จำนวน	ปริมาตรรวมสินค้า [ลบ.ซม.]	ขนส่งด้วยรถ	ระยะทาง
Cus042	P001	85,932	4	343,728	1	

ภาพที่ ก.8 โปรแกรมหน้า Worksheet: Result แสดงผลลัพธ์ที่ได้จากการคำนวณ

ภาคผนวก ข
ชุดคำสั่งโปรแกรมการจัดเส้นทางเดินรถ

มหาวิทยาลัยราชภัฏ RAJAPRUK UNIVERSITY

โปรแกรมจัดเส้นทางเดินรถถูกเขียนขึ้นบนชุดคำสั่ง VBA โดยในตัวโปรแกรมซึ่งประกอบไปด้วย Modules ส่วนต่างๆ ดังนี้

1. Modules General

```
'*****************************************************
    'เป็น โมดุล ฟังก์ชั่น สำหรับทำงานทั่วไป
'*****************************************************
Public Function Min(V1 As Double, V2 As Double) As Double
   If (V1 <= V2) Then
      Min = V1
   Else
      Min = V2
   End If
End Function
Function ConvertToLetter(iCol As Long) As String   'แปลง เลข Column เป็น ตัวอักษร
'-----------------------------------------------------------------------
' Procedure : ConvertToLetter
' Author    : Niall - Zypher.co.uk
' Purpose   : Convert a number to a column letter
'             Tested to 256 columns, assumed to work until 702 (ZZ)
'-----------------------------------------------------------------------
'
   If iCol <= 26 Then
      ' Columns A-Z
      ConvertToLetter = Chr(iCol + 64)
   Else
      ConvertToLetter = Chr(Int((iCol - 1) / 26) + 64) & _
                        Chr(((iCol - 1) Mod 26) + 65)
   End If
End Function
```

2. Modules Google

```
Public Sub DistanceBetweenPlace()  'ระยะระหว่างจุด ด้วย google Map
```

```
    Dim NumPlace As Long
    Dim Latitude_Longitude() As String
    Dim iIndex As Long
    Dim rCustomer As Range, rDistance As Range
    Dim start As Long, FinisH As Long
    Dim str1 As String, str2 As String
    Dim sDistanceAndTime As String, sDistance As String, dDistance As Double
    Dim lTemp As Long, sTemp As String
    Call SetExcelGlobalData
  Application.ScreenUpdating = False     ' ไม่ต้องให้ excel ปรับปรุงหน้าจอ เพื่อเพิ่ม
  ความเร็วในการคำนวน
    wsCalMain.Range("B9").Formula = "=COUNTA('Customer'!F10:F10000)"
    wsCalMain.Range("B9").value = wsCalMain.Range("B9").Value2
    NumPlace = wsCalMain.Range("B9").Value2   'จำนวนสถานที่
    ReDim Latitude_Longitude(1 To NumPlace) As String      'เตรียมขนาด Array
ของ Latitude Longitude

    Set rCustomer = Sheets("Customer").Range("H9")  ' กำหนดตำแหน่งอ้างอิง  ว่าอยู่
ใน Worksheeet "Customer " cell H9
    Set rDistance = Sheets("FromTo").Range("F10")  ' กำหนดตำแหน่งอ้างอิง  ว่าอยู่ใน
Worksheeet "FromTo " cell F10

    ' อ่านค่า Latitude Longitude  เก็บไว้ใน Array
    For iIndex = 1 To NumPlace
        Latitude_Longitude(iIndex) = rCustomer.Offset(iIndex, 0).value & "," &
rCustomer.Offset(iIndex, 1).value
    Next iIndex

    resave = 0

    For start = 1 To NumPlace
```

```
            For FinisH = 1 To NumPlace
                If start = FinisH Then
                    rDistance.Offset(start, FinisH).value = 0
                ElseIf rDistance.Offset(start, FinisH) > 0 Then
                    ' Do nothing
                Else
                    str1 = Latitude_Longitude(start)
                    str2 = Latitude_Longitude(FinisH)

                    'sDistanceAndTime = get_dis_and_time2(str1, str2)
                    'lTemp = InStr(1, sDistanceAndTime, " | ")
                    'sDistance = Right(sDistanceAndTime,
Len(sDistanceAndTime) - lTemp - 1)
                    'rDistance.Offset(start, FinisH).value = CLng(sDistance)

                    sDistanceAndTime = get_dis_and_time2(str1, str2)
                    lTemp = InStr(1, sDistanceAndTime, " | ")
                    sDistance = Right(sDistanceAndTime,
Len(sDistanceAndTime) - lTemp - 1)
                    rDistance.Offset(start, FinisH).value = CLng(sDistance) *
0.001

                    Application.Wait (Now + (1000 * 0.00000001)) ' หน่วงเวลา 1
วินาที
                    'Call GetDistance(str1, str2, dDistance, sTemp)
                    'rDistance.Offset(start, FinisH).value = dDistance
                End If

            Next FinisH
        Next start
        Application.StatusBar = False
```

```
End Sub

Public Function get_dis_and_time2 _
( _
origin_address As String, _
destination_address As String _
)

    ' Read more about Google API's here
    'https://developers.google.com/maps/documentation/distancematrix/

    ' we will be using Google API Distance Matrix to get Time and distance
between two cities.
    'https://developers.google.com/maps/documentation/distancematrix/

    Dim surl              As String
    Dim oXH               As Object
    Dim bodytxt           As String
    Dim tim_e             As String
    Dim distanc_e         As String

    surl = "http://maps.googleapis.com/maps/api/distancematrix/xml?origins=" & _
    Replace(Replace(Replace(origin_address, " ", "+"), ",", "+"), "++", "+") & _
    "&destinations=" & _
    Replace(Replace(Replace(destination_address, " ", "+"), ",", "+"), "++", "+") & _
    "&mode=driving&sensor=false&units=metric"

    ' units=imperial
    'if u want to show distance in kms change unit to metric
```

```
'https://developers.google.com/maps/documentation/distancematrix/#unit_systems
    'units=metric (default) returns distances in kilometers and meters.
    'units=imperial returns distances in miles and feet.

    Set oXH = CreateObject("msxml2.xmlhttp")

        With oXH
            .Open "get", surl, False
            .send
            bodytxt = .responseText
        End With

    bodytxt = Right(bodytxt, Len(bodytxt) - InStr(1, bodytxt, "<value>") - 6)
    tim_e = Left(bodytxt, InStr(1, bodytxt, "</value>") - 1)


    bodytxt = Right(bodytxt, Len(bodytxt) - InStr(1, bodytxt, "<value>") - 6)
    distanc_e = Left(bodytxt, InStr(1, bodytxt, "</value>") - 1)

    get_dis_and_time2 = tim_e & " | " & distanc_e


    Set oXH = Nothing

End Function
```

3. Modules Main

```
Option Explicit
```

```
Global wsFromTo As Worksheet, wsCustomer As Worksheet, wsProduct As
Worksheet, wsCalMain As Worksheet, wsResult As Worksheet, wsDO As Worksheet

Public ArrayOfDistance() As Double  'Array ของระยะทาง เป็น Array 2 มิติ  เริ่มต้นจาก
Index 0 ถึง City_Quant (มี Depot เป็น Index 0)

Public Sub SetExcelGlobalData()

    ' ตั้งค่าตัวแปร Work Sheet ด้วยชื่อ
    Set wsFromTo = ActiveWorkbook.Worksheets("FromTo")
    Set wsCustomer = ActiveWorkbook.Worksheets("Customer")
    Set wsProduct = ActiveWorkbook.Worksheets("Product")
    Set wsCalMain = ActiveWorkbook.Worksheets("Cal Main")
    Set wsResult = ActiveWorkbook.Worksheets("Result")

    Application.ScreenUpdating = False  'ไม่ Update หน้าจนกว่างานเสร็จ
    Application.DisplayAlerts = False  'ไม่แสดงการเตือน
End Sub

Public Sub PrepareOrderTemplate()
    Dim CustomerAllQty As Long  'จำนวน Customer
    Dim ProductAllQty As Long  'จำนวน Product

    Call SetExcelGlobalData

    'นับจำนวน Customer ทั้งหมด
    wsCalMain.Range("B9").Formula = "=COUNTA('Customer'!F11:F10000)"
    wsCalMain.Range("B9").value = wsCalMain.Range("B9").Value2
    CustomerAllQty = wsCalMain.Range("B9").Value2
    'จัดทำ Drop Down List  ให้ผู้ใช้เลือก รหัสลูกค้า
    wsCalMain.Activate
```

```
        wsCalMain.Range("K11").Select
        With Selection.Validation
            .Delete
            .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
            xlBetween, Formula1:="='Customer'!$F$11:$F$" & CStr(10 +
CustomerAllQty)
        End With
        wsCalMain.Range("K11").value = ""
        'แสดงชื่อลูกค้า
        wsCalMain.Range("L11").Formula = "=VLOOKUP(K11,Customer!$F$11:$G$" &
CStr(10 + CustomerAllQty) & ",2)"

        'นับจำนวน Product ทั้งหมด
        wsCalMain.Range("B10").Formula = "=COUNTA('Product'!F11:F10000)"
        wsCalMain.Range("B10").value = wsCalMain.Range("B10").Value2
        ProductAllQty = wsCalMain.Range("B10").Value2
        'จัดทำ Drop Down List ให้ผู้ใช้เลือก รหัส Product
        wsCalMain.Range("M11").Select
        With Selection.Validation
            .Delete
            .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
            xlBetween, Formula1:="='Product'!$F$11:$F$" & CStr(10 + ProductAllQty)
        End With
        wsCalMain.Range("M11").value = ""
        'แสดง กว้าง ยาว สูง ปริมาตร
        wsCalMain.Range("N11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",2)"
        wsCalMain.Range("O11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",3)"
        wsCalMain.Range("P11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",4)"
```

```
        wsCalMain.Range("Q11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",5)"
        wsCalMain.Range("R11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",6)"
        wsCalMain.Range("S11").Formula = "=Q11*R11"
End Sub

Private Sub ArrangeSummaryOrder(ByRef OrderAllQty As Long)   'รวมยอด order ที่
ส่งลูกค้า เจ้าเดียวกัน และ เป็นผลิตภัณฑ์อันเดียวกัน
        Dim OrderNonDupQty As Long 'จำนวน order ที่ส่งลูกค้า เจ้าเดียวกัน และ เป็น
ผลิตภัณฑ์อันเดียวกัน
        Dim CustomerID_1  As String, ProductID_1 As String, CustomerID_2  As String,
ProductID_2 As String

        Dim i As Long, j As Long

        Dim rwsCalMain As Range 'อ้างอิง Cell ใน Sheet Order
        Dim rwsResult As Range

        'นับจำนวน Order ตั้งต้น ทั้งหมด
        wsResult.Range("B10").Formula = "=COUNTA('Cal Main'!S11:S10000)"
        wsResult.Range("B10").value = wsResult.Range("B10").Value2
        OrderAllQty = wsResult.Range("B10").Value2

        wsCalMain.Range("N9").Formula = "=SUM(T11:T" &
wsResult.Range("B10").Value2 + 10 & ")"

        wsResult.Range("F11:k10000").Delete Shift:=xlUp  'ลบข้อมูลเดิม
        'คัดลอกข้อมูลจาก Sheet Order
        wsResult.Range("F11:F" & CStr(10 + OrderAllQty)).value =
wsCalMain.Range("K11:K" & CStr(10 + OrderAllQty)).value
```

```
        wsResult.Range("G11:G" & CStr(10 + OrderAllQty)).value =
wsCalMain.Range("M11:M" & CStr(10 + OrderAllQty)).value
        wsResult.Range("H11:H" & CStr(10 + OrderAllQty)).value =
wsCalMain.Range("R11:R" & CStr(10 + OrderAllQty)).value
        'ลบข้อมูลซ้ำออก
        wsResult.Range("$F$10:$H$" & CStr(10 + OrderAllQty)).RemoveDuplicates
Columns:=Array(1, 2), Header:=xlYes
        'นับจำนวน order ที่ส่งลูกค้า เจ้าเดียวกัน และ เป็นผลิตภัณฑ์อันเดียวกัน
        wsResult.Range("B11").Formula = "=COUNTA(F11:F10000)"
        wsResult.Range("B11").value = wsResult.Range("B11").Value2
        OrderNonDupQty = wsResult.Range("B11").Value2

        'เริ่มใส่ข้อมูลไม่ซ้ำ
        Set rwsCalMain = wsCalMain.Range("J10")
        Set rwsResult = wsResult.Range("F10")
        For i = 1 To OrderAllQty   'รวมจำนวน order ที่ส่งลูกค้า เจ้าเดียวกัน และ เป็น
ผลิตภัณฑ์อันเดียวกัน
            CustomerID_1 = rwsCalMain.Offset(i, 1).value: ProductID_1 =
rwsCalMain.Offset(i, 3).value
            For j = 1 To OrderNonDupQty
                CustomerID_2 = rwsResult.Offset(j, 0).value: ProductID_2 =
rwsResult.Offset(j, 1).value
                If (CustomerID_1 = CustomerID_2) And (ProductID_1 = ProductID_2)
Then
                    rwsResult.Offset(j, 3).value = rwsResult.Offset(j, 3).value +
rwsCalMain.Offset(i, 9).value
                    Exit For
                End If
            Next j
        Next i
```

```
        For j = 1 To OrderNonDupQty 'คำนวนปริมาตร
            rwsResult.Offset(j, 4).value = rwsResult.Offset(j, 2).value *
rwsResult.Offset(j, 3).value
            rwsResult.Offset(j, 5).value = rwsResult.Offset(j, 4).value
        Next j

        ' เรียงจาก Customer ID แล้วตามด้วย ปริมาตร มาก ไป น้อย
        ActiveWorkbook.Worksheets("Result").Sort.SortFields.Clear
        ActiveWorkbook.Worksheets("Result").Sort.SortFields.Add Key:=Range( _
                "F11:F" & CStr(10 + OrderNonDupQty)), SortOn:=xlSortOnValues,
Order:=xlAscending, DataOption:=xlSortNormal  '"Worksheet Result" กำหนดให้
รหัสสินค้า Column F เป็น Index
        ActiveWorkbook.Worksheets("Result").Sort.SortFields.Add Key:=Range( _
                "K11:K" & CStr(10 + OrderNonDupQty)), SortOn:=xlSortOnValues,
Order:=xlDescending, DataOption:=xlSortNormal  '"Worksheet Result" กำหนดให้
รหัสสินค้า Column K เป็น Index
        With ActiveWorkbook.Worksheets("Result").Sort  ' เรียงข้อมูลสินค้าทั้งหมด ตั้งแต่
แถวที่ 11
                .SetRange Range("F11:K" & CStr(10 + OrderNonDupQty))
                .Header = xlGuess
                .MatchCase = False
                .Orientation = xlTopToBottom
                .SortMethod = xlPinYin
                .Apply
        End With
End Sub

Private Sub FullCarLoad(maxCarLoader As Double)   'จัดเต็มรถ
    Dim OrderNonDupQty As Long 'จำนวน order ที่ส่งลูกค้า เจ้าเดียวกัน และ เป็น
ผลิตภัณฑ์อันเดียวกัน
```

```
    Dim RowProductReadNow As Long 'จำนวนแถวที่อ่าน
    Dim CarUseNow As Long 'จำนวนรถที่ใช้
    Dim RowWriteUseNow As Long 'จำนวนแถวที่เขียน

    Dim rOrderNonDup As Range 'อ้างอิง รวมยอด order ที่ส่งลูกค้า เจ้าเดียวกัน และ เป็นผลิตภัณฑ์อันเดียวกัน
    Dim rOrderFullCar_1Customer_1Product As Range 'อ้างอิง จัดแบบเต็มเที่ยว ผลิตภัณฑ์ชนิดเดียว
    Dim rOrderFullCar_1Customer_manyProduct As Range 'อ้างอิง จัดแบบเต็มเที่ยว ผลิตภัณฑ์ชนิดเดียว


    Dim bCanLoad As Boolean

    Dim BoxMaxQtyForCar As Long  'จำนวนกล่องสูงสุดที่จัดใส่รถได้

    Dim BoxRemainQty As Long  'จำนวนกล่องที่เหลือ
    Dim BoxLoadQty As Long, BoxLoadVolumn As Double  'กล่องที่ ยกขึ้นรถได้, ปริมาตรที่ ยกขึ้นรถได้
    Dim TotalCustomerVolumn As Double  ' ปริมาตรสินค้าของลูกค้า
    Dim BoxSize As Double 'ปริมาตรกล่อง

    Dim ContainerReaminV As Double  'ปริมาตรในตู้ที่คงเหลือ

    Dim CustomerQty As Long  'จำนวนลูกค้า

    Dim i As Long, indexProduct As Long

    Dim rRemainProduct_ByCustomer As Range   'อ้างอิง ปริมาตร คงเหลือ รวม

    Dim DistanceTotal As Double 'ระยะทาง
```

```
    OrderNonDupQty = wsResult.Range("B11").Value2

    Set rOrderNonDup = wsResult.Range("F10")
    Set rOrderFullCar_1Customer_1Product = wsResult.Range("O10")

     '*************************************************
          'จัดแบบ Product เดียวเต็มคันก่อน
     '*************************************************

    wsResult.Range("N11:U10000").Delete Shift:=xlUp  'ลบข้อมูลเดิม
    CarUseNow = 0: RowWriteUseNow = 0  "จำนวนรถที่ใช้, จำนวนแถวที่เขียน
    RowProductReadNow = 1
    Do
        bCanLoad = False
        For i = RowProductReadNow To OrderNonDupQty
            BoxSize = rOrderNonDup.Offset(i, 2).value
            BoxMaxQtyForCar = Int(maxCarLoader / BoxSize)

            If BoxMaxQtyForCar <= Round(rOrderNonDup.Offset(i, 5).value / BoxSize,
0) Then 'บรรจุได้เต็มคัน

                rOrderNonDup.Offset(i, 5).value = rOrderNonDup.Offset(i, 5).value -
BoxMaxQtyForCar * BoxSize  'หักปริมาณที่ส่งเต็มรถ

                CarUseNow = CarUseNow + 1: RowWriteUseNow = RowWriteUseNow
+ 1  'เพิ่มเลขรถที่ใช้

                 rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, -1).value
= NameCustomerToIndex(rOrderNonDup.Offset(i, 0).value)
```

```
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, 0).value
= rOrderNonDup.Offset(i, 0).value
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, 1).value
= rOrderNonDup.Offset(i, 1).value
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, 2).value
= BoxSize
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow,
2).NumberFormat = "#,##0"
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, 3).value
= BoxMaxQtyForCar
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, 4).value
= BoxMaxQtyForCar * BoxSize
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow,
4).NumberFormat = "#,##0"
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, 5).value
= CarUseNow


                wsFromTo.Range("I3").value = 0
                wsFromTo.Range("I4").value =
rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, -1).value
                DistanceTotal = wsFromTo.Range("I5").value     'ระยะทางขาไป
                'รวมระยะทางขากลับ
                wsFromTo.Range("I4").value = 0: wsFromTo.Range("I3").value =
rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, -1).value
                DistanceTotal = DistanceTotal + wsFromTo.Range("I5").value
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow, 6).value
= DistanceTotal
                rOrderFullCar_1Customer_1Product.Offset(RowWriteUseNow,
6).NumberFormat = "#,##0.00"
```

```
                If rOrderNonDup.Offset(i, 5).value < maxCarLoader Then  'บรรจุได้ไม่เต็มคัน  ไปอ่านบรรทัด ถัดไป
                    RowProductReadNow = RowProductReadNow + 1
                End If
                bCanLoad = True   'จัดรถได้
                Exit For
            End If
        Next i
    Loop Until (bCanLoad = False)
    wsResult.Range("B17").value = CarUseNow
    wsResult.Range("C17").value = CarUseNow

    '*****************************************************
        'จัดแบบหลาย Product ลูกค้าคนเดียว
    '*****************************************************
    wsResult.Range("Y11:AB10000").Delete Shift:=xlUp  'ลบข้อมูลเดิม
    wsResult.Range("Y11:Y" & CStr(10 + OrderNonDupQty)).value =
wsResult.Range("F11:F" & CStr(10 + OrderNonDupQty)).value
    'ลบข้อมูลซ้ำออก
    wsResult.Range("$Y$10:$Y$" & CStr(10 + OrderNonDupQty)).RemoveDuplicates
Columns:=1, Header:=xlYes
    wsResult.Range("B14").Formula = "=COUNTA(Y11:Y10000)":
wsResult.Range("B14").value = wsResult.Range("B14").Value2  'นับจำนวนลูกค้า
    CustomerQty = wsResult.Range("B14").value

    'นับจำนวนผลิตภัณฑ์ที่ลูกค้าคงเหลือ
    wsResult.Range("Z11").Formula = "=SUMIF($F$11:$F$" & CStr(10 +
OrderNonDupQty) & ",Y11,$K$11:$K$" & CStr(10 + OrderNonDupQty) & ")":  'จำนวนปริมาตร Product ของลูกค้า
    wsResult.Range("AA11").Formula = "=MATCH(Y11,$F$11:$F$" & CStr(10 +
OrderNonDupQty) & ",0)"     'Index เริ่มต้นของลูกค้า
```

```
    If (CustomerQty > 1) Then
        wsResult.Range("Z11:AA11").Copy Destination:=wsResult.Range("Z12:AA" &
CStr(10 + CustomerQty))
    End If
    wsResult.Range("Z11:AA" & CStr(10 + CustomerQty)).value =
wsResult.Range("Z11:AA" & CStr(10 + CustomerQty)).Value2

    For i = 1 To CustomerQty - 1  'Index ปลายของลูกค้า
        wsResult.Range("AB" & CStr(10 + i)).value = wsResult.Range("AA" & CStr(10 + i
+ 1)).value - 1
    Next i
    wsResult.Range("AB" & CStr(10 + CustomerQty)).value = OrderNonDupQty

    Set rRemainProduct_ByCustomer = wsResult.Range("Y10")
    Set rOrderFullCar_1Customer_manyProduct = wsResult.Range("AH10")

    wsResult.Range("AG11:AN10000").Delete Shift:=xlUp  'ลบข้อมูลเดิม
    CarUseNow = 1: RowWriteUseNow = 0  "จำนวนรถที่ใช้, จำนวนแถวที่เขียน

    For i = 1 To CustomerQty   'จัดการไปทีละลูกค้า
        ContainerReaminV = maxCarLoader
        Do While (rRemainProduct_ByCustomer.Offset(i, 1).value >= maxCarLoader)
            'CarUseNow = CarUseNow + 1
            For indexProduct = rRemainProduct_ByCustomer.Offset(i, 2).value To
rRemainProduct_ByCustomer.Offset(i, 3).value
                BoxSize = rOrderNonDup.Offset(indexProduct, 2).value
                BoxLoadQty = Int(rOrderNonDup.Offset(indexProduct, 5).value /
BoxSize)  'จำนวนกล่อง สินค้าที่มี
                BoxMaxQtyForCar = Application.Min(Int(ContainerReaminV / BoxSize),
BoxLoadQty)
```

```
If BoxMaxQtyForCar > 0 Then    'จัดขึ้นรถได้

        rOrderNonDup.Offset(indexProduct, 5).value =
rOrderNonDup.Offset(indexProduct, 5).value - BoxMaxQtyForCar * BoxSize  'หัก
ปริมาณที่ส่งขึ้นรถ
        rRemainProduct_ByCustomer.Offset(i, 1).value =
rRemainProduct_ByCustomer.Offset(i, 1).value - BoxMaxQtyForCar * BoxSize  'หัก
ปริมาณที่ส่งขึ้นรถ

        RowWriteUseNow = RowWriteUseNow + 1  'เพิ่มเลขบรรทัดที่ใช้

        rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow, -
1).value = NameCustomerToIndex(rOrderNonDup.Offset(indexProduct, 0).value)
        rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
0).value = rOrderNonDup.Offset(indexProduct, 0).value
        rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
1).value = rOrderNonDup.Offset(indexProduct, 1).value
        rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
2).value = BoxSize
        rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
2).NumberFormat = "#,##0"
        rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
3).value = BoxMaxQtyForCar
        rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
4).value = BoxMaxQtyForCar * BoxSize
        rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
4).NumberFormat = "#,##0"
        rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
5).value = CarUseNow
```

```
                ContainerReaminV = ContainerReaminV - BoxMaxQtyForCar *
BoxSize

                If (ContainerReaminV < BoxSize) Then  'จัดได้เต็มรถ เพิ่มเลขรถ
                    wsFromTo.Range("I3").value = 0: wsFromTo.Range("I4").value =
rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow, -1).value
                    DistanceTotal = wsFromTo.Range("I5").value     'ระยะทางขาไป
                    'รวมระยะทางขากลับ
                    wsFromTo.Range("I4").value = 0: wsFromTo.Range("I3").value =
rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow, -1).value
                     DistanceTotal = DistanceTotal + wsFromTo.Range("I5").value
                    rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
6).value = DistanceTotal
                    rOrderFullCar_1Customer_manyProduct.Offset(RowWriteUseNow,
6).NumberFormat = "#,##0.00"
                    ContainerReaminV = maxCarLoader
                    CarUseNow = CarUseNow + 1
                End If
            End If
        Next indexProduct
    Loop
  Next i

End Sub

Private Function SummaryAndArrangrCustomerRemain() As Long  'จัดการ ลบลูกค้าที่
ปริมาตร คงเหลือ รวม = 0 แล้วคืนค่าลูกค่าที่เหลือ
  Dim i As Long
  Dim numCustomer As Long, numRow As Long
```

```
    wsResult.Range("B4").Formula = "=COUNTA(Y11:Y10000)":
wsResult.Range("B4").value = wsResult.Range("B4").Value2
    numCustomer = wsResult.Range("B4").value: numRow = numCustomer

    For i = numRow To 1 Step -1
        If Round(wsResult.Range("Z" & CStr(10 + i)).value, 0) <= 0 Then 'ค่าเป็น 0
ลบทิ้ง
            wsResult.Range("Y" & CStr(10 + i) & ":AB" & CStr(10 + i)).Delete Shift:=xlUp
'ลบข้อมูลเดิม
            numCustomer = numCustomer - 1
        End If
    Next i

    SummaryAndArrangrCustomerRemain = numCustomer
End Function

Private Sub NearestNeighborLoad(maxCarLoader As Double, maxDistance As
Double)
    Dim CustomerQty As Long
    Dim rRemainTotalProduct_ByCustomer As Range
    Dim rOrderNonDup As Range
    Dim rOrderCarLoadProduct As Range

    Dim CarUseNow As Long, RowWriteUseNow As Long
    Dim Station_Start As Long, Station_End As Long  'station เริ่ม-ปลาย
    Dim indexProduct_Start As Long, indexProduct_End As Long

    Dim Nearless_Distance As Double  'ระยะทางที่สั้นสุด
    Dim Nearless_Station As Long  'Station ระยะทางที่สั้นสุด
    Dim ContainerReaminV As Double  'พื้นที่ ที่เหลือในตู้
```

```
    Dim aCustomerIndexArray() As Long   'Index ลูกค้าที่ยังเหลือการขน
    Dim i_custom As Long
    Dim i_customer_total  As Long, i_customer_detailStart As Long,
i_customer_detailEnd As Long
    Dim rowCustomerTotal As Long, tmpLong2 As Long

    Dim bFoundBetter As Boolean  'เจอสถานีใหม่ที่สั้นกว่า

    Dim BoxRemainQty As Long  'จำนวนกล่องที่เหลือ
    Dim BoxLoadQty As Long, BoxLoadVolumn As Double  'กล่องที่ ยกขึ้นรถได้,
ปริมาตรที่ ยกขึ้นรถได้
    Dim BoxMaxQtyForCar As Long
    Dim TotalCustomerVolumn As Double  ' ปริมาตรสินค้าของลูกค้า
    Dim totalDistanceNotReturnToDepot As Double, TestDistance As Double
'ระยะทาง
    Dim CustomerInCar As Long   'จำนวนชนิดสินค้า
    Dim sConstrainBy As String   'จำกัดโดย
    Dim BoxSize As Double 'ปริมาตรกล่อง

    Dim i As Long, j As Long, m As Long, n As Long
    Dim indexProduct As Long

    Dim aStation() As Long   'ลูกค้าที่จัดส่ง

    Set rOrderNonDup = wsResult.Range("F10")
    Set rRemainTotalProduct_ByCustomer = wsResult.Range("Y10")
    Set rOrderCarLoadProduct = wsResult.Range("AS10")

    wsResult.Range("AR11:AZ10000").Delete Shift:=xlUp  'ลบข้อมูลเดิม
    CustomerQty = SummaryAndArrangrCustomerRemain()
```

```
    ReDim aCustomerIndexArray(1 To CustomerQty)      'เตรียมขนาด Array เท่ากับ
จำนวนลูกค้า
    Set rRemainTotalProduct_ByCustomer = wsResult.Range("Y10")

    wsResult.Range("Z6").Formula = "=MATCH(Y6,Y11:Y10000,0)"   'ใส่สูตรค้นหาลำดับที่
จากชื่อ
    CarUseNow = 1: RowWriteUseNow = 0  "จำนวนรถที่ใช้, จำนวนแถวที่เขียน
    Do While (CustomerQty > 0)
        ContainerReaminV = maxCarLoader
        CustomerInCar = 0

        For i_custom = 1 To CustomerQty
            aCustomerIndexArray(i_custom) =
NameCustomerToIndex(rRemainTotalProduct_ByCustomer.Offset(i_custom,
0).value)       'เริ่ม ใส่ Index ใน Array
        Next i_custom

        'เริ่มหา NearLess
        Station_Start = 0  'เริ่มที่ Depot ก่อน
        totalDistanceNotReturnToDepot = 0
        sConstrainBy = "Constrain by Volumn"  'คิดว่าจำกัดโดย Volumn ไว้ก่อน

        Do
            Nearless_Distance = 1E+99
            bFoundBetter = False 'คิดว่าหาตัวสั้นกว่าไม่เจอ

            For i = 1 To UBound(aCustomerIndexArray)
                wsResult.Range("Y6").value =
IndexCustomerToName(aCustomerIndexArray(i))
                rowCustomerTotal = wsResult.Range("Z6")
```

```
            indexProduct_Start =
rRemainTotalProduct_ByCustomer.Offset(rowCustomerTotal, 2).value
            indexProduct_End =
rRemainTotalProduct_ByCustomer.Offset(rowCustomerTotal, 3).value

            If (rRemainTotalProduct_ByCustomer.Offset(rowCustomerTotal,
1).value <= ContainerReaminV) And
(rRemainTotalProduct_ByCustomer.Offset(rowCustomerTotal, 1).value > 0) Then
                  If (ArrayOfDistance(Station_Start, aCustomerIndexArray(i)) <
Nearless_Distance) Then

                        Nearless_Station = aCustomerIndexArray(i)
                        Nearless_Distance = ArrayOfDistance(Station_Start,
aCustomerIndexArray(i))

                        bFoundBetter = True  'เจอ Station ที่สั้นกว่า
                  End If
            End If
      Next i

      If bFoundBetter = True Then  'เจอ Station ที่สั้นกว่า  เขียน Data
            totalDistanceNotReturnToDepot = totalDistanceNotReturnToDepot +
Nearless_Distance
            TestDistance = totalDistanceNotReturnToDepot +
ArrayOfDistance(Nearless_Station, 0)


            If (maxDistance >= TestDistance) Or (CustomerInCar = 0) Then   'จัดส่ง
ได้  ทั้งที่ระยะไม่เกิน และระยะเกิน
```

```
                wsResult.Range("Y6").value =
IndexCustomerToName(Nearless_Station)
                rowCustomerTotal = wsResult.Range("Z6")   'เจอแถวที่
                indexProduct_Start =
rRemainTotalProduct_ByCustomer.Offset(rowCustomerTotal, 2).value
                indexProduct_End =
rRemainTotalProduct_ByCustomer.Offset(rowCustomerTotal, 3).value

                For indexProduct = indexProduct_Start To indexProduct_End
                   BoxSize = rOrderNonDup.Offset(indexProduct, 2).value
                   BoxLoadQty = CLng(Round(rOrderNonDup.Offset(indexProduct,
5).value, 8) / BoxSize) 'จำนวนกล่อง สินค้าที่มี
                   BoxMaxQtyForCar = Application.Min(Int(ContainerReaminV /
BoxSize), BoxLoadQty)


                   If BoxMaxQtyForCar > 0 Then   'จัดขึ้นรถได้

                      rOrderNonDup.Offset(indexProduct, 5).value =
Round(rOrderNonDup.Offset(indexProduct, 5).value - BoxMaxQtyForCar * BoxSize,
6) 'หักปริมาณที่ส่งขึ้นรถ
                      rRemainTotalProduct_ByCustomer.Offset(rowCustomerTotal,
1).value = Round(rRemainTotalProduct_ByCustomer.Offset(rowCustomerTotal,
1).value - BoxMaxQtyForCar * BoxSize, 6) 'หักปริมาณที่ส่งขึ้นรถ

                      RowWriteUseNow = RowWriteUseNow + 1  'เพิ่มเลขบรรทัดที่ใช้

                      rOrderCarLoadProduct.Offset(RowWriteUseNow, -1).value =
NameCustomerToIndex(rOrderNonDup.Offset(indexProduct, 0).value)
                      rOrderCarLoadProduct.Offset(RowWriteUseNow, 0).value =
rOrderNonDup.Offset(indexProduct, 0).value
```

```
                    rOrderCarLoadProduct.Offset(RowWriteUseNow, 1).value =
rOrderNonDup.Offset(indexProduct, 1).value
                    rOrderCarLoadProduct.Offset(RowWriteUseNow, 2).value =
BoxSize
                    rOrderCarLoadProduct.Offset(RowWriteUseNow,
2).NumberFormat = "#,##0"
                    rOrderCarLoadProduct.Offset(RowWriteUseNow, 3).value =
BoxMaxQtyForCar
                    rOrderCarLoadProduct.Offset(RowWriteUseNow, 4).value =
BoxMaxQtyForCar * BoxSize
                    rOrderCarLoadProduct.Offset(RowWriteUseNow,
4).NumberFormat = "#,##0"
                    rOrderCarLoadProduct.Offset(RowWriteUseNow, 5).value =
CarUseNow

                    ContainerReaminV = ContainerReaminV - BoxMaxQtyForCar *
BoxSize
                Else

                End If
            Next indexProduct

            CustomerInCar = CustomerInCar + 1

            Station_Start = Nearless_Station     'จุดปลาย เป็นจุดเริ่ม
        End If
    End If

    If (maxDistance < TestDistance) Then   'จัดส่งได้  ทั้งที่ระยะเกิน
        sConstrainBy = "Constrain by Distance"
        bFoundBetter = False
```

```
            End If

        Loop Until bFoundBetter = False

        rOrderCarLoadProduct.Offset(RowWriteUseNow, 7).value = sConstrainBy

        'จัดได้เต็มรถ เพิ่มเลขรถ
        CustomerQty = SummaryAndArrangrCustomerRemain()
        If CustomerQty > 0 Then
            ReDim aCustomerIndexArray(1 To CustomerQty)
        End If
        ContainerReaminV = maxCarLoader
        CarUseNow = CarUseNow + 1
    Loop
    wsResult.Range("B19").value = CarUseNow - 1   'จำนวนรถ
    wsResult.Range("C19").value = RowWriteUseNow
     wsResult.Range("Z6").ClearContents    'ลบสูตรออกลดการคำนวน

     For i = 1 To CarUseNow - 1
        RowWriteUseNow = ReadStationInCar("AX", "AR", i, aStation)
        rOrderCarLoadProduct.Offset(RowWriteUseNow - 10, 6).value =
Tour_Distance(aStation)
        rOrderCarLoadProduct.Offset(RowWriteUseNow - 10, 6).NumberFormat =
"#,##0.00"
        'rOrderCarLoadProduct.Offset(RowWriteUseNow, 6).value
     Next i
End Sub

Public Sub FindSolution()
    Dim maxCarLoader As Double  'ปริมาตรรถสูงสุดที่บรรทุกได้
    Dim maxDistance As Double      'ระยะทางสูงสุดที่เดินทาง
```

```
    Dim OrderAllQty As Long  'จำนวน Order ตั้งต้น ทั้งหมด
    Dim City_Quant As Long
    Dim ResultArray As Variant
    Dim time1 As Double
    Dim time2 As Double

    Dim rDistanceMatrix As Range
    Dim i As Long, j As Long

    Call SetExcelGlobalData
    Call ArrangeSummaryOrder(OrderAllQty)   'รวมยอด order ที่ส่งลูกค้า เจ้าเดียวกัน
และ เป็นผลิตภัณฑ์อันเดียวกัน

    wsResult.Range("B17:C19").ClearContents   'ล้างข้อมูลสรุปจำนวนรถ

    maxCarLoader = wsCalMain.Range("P6").value
    maxDistance = wsCalMain.Range("Q6").value

    City_Quant =
Worksheets("FromTo").Range("G12:G10000").Cells.SpecialCells(xlCellTypeConstants)
.Count
    Set rDistanceMatrix = wsFromTo.Range("G11")

    ReDim ArrayOfDistance(0 To City_Quant, 0 To City_Quant)
      For i = 0 To City_Quant   'ใส่ข้อมูลจาก Depot ถึงลูกค้า
        For j = 0 To City_Quant
                ArrayOfDistance(i, j) = rDistanceMatrix.Offset(i, j).value
        Next j
    Next i
```

```
    Call FullCarLoad(maxCarLoader)
    'คำนวน ระยะทาง ขนเต็มเที่ยว Product เดียว
    wsResult.Range("U8").Formula = "=SUM(U11:U10000)":
wsResult.Range("U8").value = wsResult.Range("U8").value
    wsResult.Range("U8").NumberFormat = "#,##0.00"
    'คำนวน ระยะทาง ขนเต็มเที่ยวหลาย Product
    wsResult.Range("AN8").Formula = "=SUM(AN11:AN10000)":
wsResult.Range("AN8").value = wsResult.Range("AN8").value
    wsResult.Range("AN8").NumberFormat = "#,##0.00"
    Call NearestNeighborLoad(maxCarLoader, maxDistance)
    'คำนวน ระยะทาง
    wsResult.Range("AY8").Formula = "=SUM(AY11:AY10000)":
wsResult.Range("AY8").value = wsResult.Range("AY8").value
    wsResult.Range("AY8").NumberFormat = "#,##0.00"
    time1 = Timer
    Call Cal3OptLoad
    'คำนวน ระยะทาง
    wsResult.Range("BK8").Formula = "=SUM(BK11:BK10000)":
wsResult.Range("BK8").value = wsResult.Range("BK8").value
    wsResult.Range("BK8").NumberFormat = "#,##0.00"
    time2 = Timer
    wsCalMain.Range("R4").value = time2 - time1

    time1 = Timer
    Call CalSALoad
    'คำนวน ระยะทาง
    wsResult.Range("BW8").Formula = "=SUM(BW11:BW10000)":
wsResult.Range("BW8").value = wsResult.Range("BW8").value
    wsResult.Range("BW8").NumberFormat = "#,##0.00"
    time2 = Timer
```

```
    wsCalMain.Range("S4").value = time2 - time1

End Sub
Private Function IndexCustomerToName(indexCustomer As Long) As String   'ใส่
Index ลูกค้า แปลงเป็นชื่อ
    wsFromTo.Range("N3").value = indexCustomer
    IndexCustomerToName = wsFromTo.Range("M3").value
End Function

Private Function NameCustomerToIndex(CustomerName As String) As Long   'ใส่
ชื่อลูกค้า แปลงเป็น Index
    wsFromTo.Range("M4").value = CustomerName
    NameCustomerToIndex = wsFromTo.Range("N4").value
End Function

'Private Function ArrayOfDistance(StartIndex As Long, EndIndex As Long) As
Double
    'wsFromTo.Range("I3").value = StartIndex
    'wsFromTo.Range("I4").value = EndIndex
   ' ArrayOfDistance = wsFromTo.Range("I5").value
'End Function

Private Function DistanceByName(StartStationName As String, EndStationName As
String) As Double
    wsFromTo.Range("F3").value = StartIndex
    wsFromTo.Range("F4").value = EndIndex
    ArrayOfDistance = wsFromTo.Range("F5").value
End Function

Public Function Tour_Distance(ArrayTour As Variant) As Double   'ระยะทางทั้งหมด
วนรอบจนกลับ Depot
```

```
    Dim LB As Long, UB As Long, i As Long
    Dim BeginStation As Long, totalDistance As Double


    LB = LBound(ArrayTour): UB = UBound(ArrayTour)   'ขอบเขต ของ Array

     totalDistance = 0: BeginStation = 0
    If LB <= UB Then
        For i = LB To UB
            totalDistance = totalDistance + ArrayOfDistance(BeginStation,
CLng(ArrayTour(i)))
            BeginStation = ArrayTour(i)
        Next i
        totalDistance = totalDistance + ArrayOfDistance(BeginStation, 0)
    End If

    Tour_Distance = totalDistance
End Function

Public Function ReadStationInCar(sCar As String, sStationID As String, carNo As
Long, ResultArray As Variant) As Long 'อ่านค่าสถานีจากแถวที่กำหนด คืนค่าแถวสุดท้าย
    Dim row As Long, rowFinal As Long
    Dim stationOld As Long
    Dim numStation As Long

    row = 11: numStation = 0: stationOld = 100000
    Do While CLng(wsResult.Range(sCar & CStr(row)).value) <> 0
        If wsResult.Range(sCar & CStr(row)).value = carNo Then
            If wsResult.Range(sStationID & CStr(row)).value <> stationOld Then
                numStation = numStation + 1
                stationOld = wsResult.Range(sStationID & CStr(row)).value
```

```
            End If
        Else
            If (wsResult.Range(sCar & CStr(row)).value > carNo) Then    'Or
(CLng(wsResult.Range(sCar & CStr(row)).value) = 0)
                Exit Do
            End If
        End If
        row = row + 1
    Loop

    rowFinal = row - 1

    ReDim ResultArray(1 To numStation)
    row = 11: numStation = 0: stationOld = 100000
    For row = 11 To rowFinal
        If wsResult.Range(sCar & CStr(row)).value = carNo And
wsResult.Range(sStationID & CStr(row)).value <> stationOld Then
            numStation = numStation + 1
            ResultArray(numStation) = wsResult.Range(sStationID & CStr(row)).value
            stationOld = wsResult.Range(sStationID & CStr(row)).value
        End If
    Next row

    ReadStationInCar = rowFinal

End Function

Public Sub PrepareDataInputTemplate()
        Dim CustomerAllQty As Long  'จำนวน Customer
        Dim ProductAllQty As Long  'จำนวน Product
```

```
        Call SetExcelGlobalData

        'นับจำนวน Customer ทั้งหมด
        wsCalMain.Range("B9").Formula = "=COUNTA('Customer'!F11:F10000)"
        wsCalMain.Range("B9").value = wsCalMain.Range("B9").Value2
        CustomerAllQty = wsCalMain.Range("B9").Value2
        'จัดทำ Drop Down List  ให้ผู้ใช้เลือก รหัสลูกค้า
        'wsCalMain.Activate
        wsCalMain.Range("K11").Select
        With Selection.Validation
            .Delete
            .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
            xlBetween, Formula1:="='Customer'!$F$11:$F$" & CStr(10 +
CustomerAllQty)
        End With
       ' wsCalMain.Range("C1").value = ""
        'แสดงชื่อลูกค้า
        wsCalMain.Range("L11").Formula = "=VLOOKUP(K11,Customer!$F$11:$G$" &
CStr(10 + CustomerAllQty) & ",2)"

        'นับจำนวน Product ทั้งหมด
        wsCalMain.Range("B10").Formula = "=COUNTA('Product'!F11:F10000)"
        wsCalMain.Range("B10").value = wsCalMain.Range("B10").Value2
        ProductAllQty = wsCalMain.Range("B10").Value2
        'จัดทำ Drop Down List  ให้ผู้ใช้เลือก รหัส Product
        wsCalMain.Range("M11").Select
        With Selection.Validation
            .Delete
            .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
            xlBetween, Formula1:="='Product'!$F$11:$F$" & CStr(10 + ProductAllQty)
        End With
```

```
        'แสดง  ชื่อผลิตภัณฑ์  กว้าง   ยาว สูง ปริมาตร
        wsCalMain.Range("N11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",2)"
        wsCalMain.Range("O11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",3)"
        wsCalMain.Range("P11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",4)"
        wsCalMain.Range("Q11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",5)"
        wsCalMain.Range("R11").Formula = "=VLOOKUP(M11,Product!$F$11:$K$" &
CStr(10 + ProductAllQty) & ",6)"
        wsCalMain.Range("T11").Formula = "=R11*S11"

        'ล้างค่า
        wsCalMain.Range("K11").value = ""
        wsCalMain.Range("M11").value = ""
        wsCalMain.Range("S11").value = ""

        wsCalMain.Range("K11:T11").Copy
Destination:=wsCalMain.Range("K12:K1010")
End Sub

Public Sub CreateFromToLabel()
    Dim CustomerAllQty As Long   'จำนวนลูกค้าไม่รวม Depot
    Dim DestinationQty As Long   'จำนวนลูกค้ารวม Depot
    Dim tmpColName As String  'ชื่อ Col แบบตัวอักษร

    Call SetExcelGlobalData
    'นับจำนวน Customer ทั้งหมด
        wsCalMain.Range("B9").Formula = "=COUNTA('Customer'!F11:F10000)"
```

```
        wsCalMain.Range("B9").value = wsCalMain.Range("B9").Value2
        CustomerAllQty = wsCalMain.Range("B9").Value2

        DestinationQty = CustomerAllQty + 1

        wsFromTo.Range("F12:SS10000").Clear
        wsFromTo.Range("H10:SS11").Clear
        wsFromTo.Range("F10:G11").ClearContents

        wsFromTo.Range("F11").Copy Destination:=wsFromTo.Range("F" & CStr(11) &
":F" & CStr(10 + DestinationQty))
        tmpColName = ConvertToLetter(6 + DestinationQty)
        wsFromTo.Range("F11").Copy Destination:=wsFromTo.Range("G10:" &
tmpColName & CStr(10))

        wsFromTo.Range("G11").Copy Destination:=wsFromTo.Range("G11:" &
tmpColName & CStr(10 + DestinationQty))

        wsFromTo.Range("F11:F" & CStr(10 + DestinationQty)).value =
wsCustomer.Range("F10:F" & CStr(10 + DestinationQty - 1)).value
        wsFromTo.Range("G10:" & tmpColName & CStr(10)).value =
Application.Transpose(wsCustomer.Range("F10:F" & CStr(10 + DestinationQty -
1)).value)
End Sub
```

4. Modules Metrix

```
Option Explicit

Public Const C_ERR_NO_ERROR = 0&
Public Const C_ERR_SUBSCRIPT_OUT_OF_RANGE = 9&
Public Const C_ERR_ARRAY_IS_FIXED_OR_LOCKED = 10&
```

```
Public Sub ArrayDeleteByIndex(ArrayOrg As Variant, value As Long)    'ลบสมาชิก
Array จากค่าที่กำหนด
    Dim LB As Long, UB As Long, indexFound As Long
    Dim ArrayTemp() As Long
    Dim found As Boolean
    Dim i As Long

    LB = LBound(ArrayOrg): UB = UBound(ArrayOrg)   'ขอบเขต ของ Array

    If LB < UB Then
        ReDim ArrayTemp(LB To UB)
        For i = LB To UB 'คัดลอกค่ามายัง   ArrayTemp
            ArrayTemp(i) = ArrayOrg(i)
        Next i

        ReDim ArrayOrg(LB To (UB - 1))    'ลดค่า ArrayOrg ลง 1

        i = LB: found = False
        Do Until found = True
            If (ArrayTemp(i) = value) Then
                indexFound = i
                found = True
            Else
                ArrayOrg(i) = ArrayTemp(i)
            End If
            i = i + 1
        Loop

        For i = indexFound To UB - 1
            ArrayOrg(i) = ArrayTemp(i + 1)
```

```
            Next i
        Else
            ReDim ArrayOrg(0)
        End If
End Sub

Public Sub ArrayInsertByIndex(ArrayOrg As Variant, Index As Long, value As Long)
'เพิ่มสมาชิก Array จากค่าที่กำหนด
        Dim LB As Long, UB As Long, indexFound As Long
        Dim ArrayTemp() As Long
        Dim found As Boolean
        Dim i As Long

        LB = LBound(ArrayOrg): UB = UBound(ArrayOrg)   'ขอบเขต ของ Array

        If LB < UB Then
            ReDim ArrayTemp(LB To UB)
            For i = LB To UB 'คัดลอกค่ามายัง   ArrayTemp
                ArrayTemp(i) = ArrayOrg(i)
            Next i

            ReDim ArrayOrg(LB To (UB + 1))    'เพิ่มค่า ArrayOrg ขึ้น 1
            ArrayOrg(Index) = value

            For i = LB To Index - 1
               ArrayOrg(i) = ArrayTemp(i)
            Next i

            For i = Index + 1 To UB + 1
               ArrayOrg(i) = ArrayTemp(i - 1)
            Next i
```

```
        Else
                ReDim ArrayOrg(0)
        End If
End Sub

Public Sub Array_Copy(OriginArray As Variant, TargetArray As Variant)   'สำหรับ
Copy Array
        Dim LB As Long, UB As Long   'ขนาด Array
        Dim i As Integer

        LB = LBound(OriginArray)
        UB = UBound(OriginArray)    'อ่านขนาด Array
        ReDim TargetArray(LB To UB)
        For i = LB To UB
                TargetArray(i) = OriginArray(i)
        Next i
End Sub

Public Sub SwapBit(CustomerArray As Variant, ByVal StartIndex As Long, ByVal
EndIndex As Long)      ' สลับแบบ  Opt
        Dim i As Long, tmpLong As Long, tmpIndexLower As Long

        For i = 1 To ((EndIndex - StartIndex + 1) \ 2)
                tmpIndexLower = StartIndex + i - 1
                tmpLong = CustomerArray(tmpIndexLower)
                CustomerArray(tmpIndexLower) = CustomerArray(EndIndex)
                CustomerArray(EndIndex) = tmpLong
                EndIndex = EndIndex - 1
        Next i
End Sub
```

```
Public Function FindStationInArray(ArrayOrg As Variant, valueSearch As Long) As
Long    'คืนตำแหน่ว Array จาก Station ที่กำหนด
        Dim LB As Long, UB As Long, indexFound As Long
        Dim i As Long

        LB = LBound(ArrayOrg): UB = UBound(ArrayOrg)   'ขอบเขต ของ Array

        If LB <= UB Then
                For i = LB To UB 'คัดลอกค่ามายัง   ArrayTemp
                        If valueSearch = ArrayOrg(i) Then
                            FindStationInArray = i
                            Exit Function
                        End If
                Next i
        End If
        FindStationInArray = 0
End Function

Public Function CopyArraySubSetToArray(InputArray As Variant, ResultArray As
Variant, _
    FirstElementToCopy As Long, LastElementToCopy As Long, DestinationElement
As Long) As Boolean
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' CopyArraySubSetToArray
' This function copies elements of InputArray to ResultArray. It takes the elements
' from FirstElementToCopy to LastElementToCopy (inclusive) from InputArray and
' copies them to ResultArray, starting at DestinationElement. Existing data in
' ResultArray will be overwrittten. If ResultArray is a dynamic array, it will
' be resized if needed. If ResultArray is a static array and it is not large
' enough to copy all the elements, no elements are copied and the function
```

```
' returns False.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Dim SrcNdx As Long
Dim DestNdx As Long
Dim NumElementsToCopy As Long

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Set the default return value.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
CopyArraySubSetToArray = False

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Ensure InputArray and ResultArray are
' arrays.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If IsArray(InputArray) = False Then
    Exit Function
End If
If IsArray(ResultArray) = False Then
    Exit Function
End If
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Ensure InputArray is single dimensional.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
If NumberOfArrayDimensions(Arr:=InputArray) <> 1 Then
    Exit Function
End If
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Ensure ResultArray is unallocated or
' single dimensional.
```

```
''''''''''''''''''''''''''''''''''''''''''''''
If NumberOfArrayDimensions(Arr:=ResultArray) > 1 Then
    Exit Function
End If

''''''''''''''''''''''''''''''''''''''''''''''
' Ensure the bounds and indexes are valid.
''''''''''''''''''''''''''''''''''''''''''''''
If FirstElementToCopy < LBound(InputArray) Then
    Exit Function
End If
If LastElementToCopy > UBound(InputArray) Then
   Exit Function
End If
If FirstElementToCopy > LastElementToCopy Then
    Exit Function
End If

''''''''''''''''''''''''''''''''''''''''''''
' Calc the number of elements we'll copy
' from InputArray to ResultArray.
''''''''''''''''''''''''''''''''''''''''''''
NumElementsToCopy = LastElementToCopy - FirstElementToCopy + 1

If IsArrayDynamic(Arr:=ResultArray) = False Then
    If (DestinationElement + NumElementsToCopy - 1) > UBound(ResultArray)
Then
        ''''''''''''''''''''''''''''''''''''''''''''''''
        ' ResultArray is static and can't be resized.
        ' There is not enough room in the array to
        ' copy all the data.
```

```
        ''''''''''''''''''''''''''''''''''''''''
        Exit Function
    End If
Else
    ''''''''''''''''''''''''''''''''''''''''
    ' ResultArray is dynamic and can be resized.
    ' Test whether we need to resize the array,
    ' and resize it if required.
    ''''''''''''''''''''''''''''''''''''''''
    If IsArrayEmpty(Arr:=ResultArray) = True Then
        ''''''''''''''''''''''''''''''''''''
        ' ResultArray is unallocated. Resize it
        ' to DestinationElement + NumElementsToCopy - 1.
        ' This provides empty elements to the left
        ' of the DestinationElement and room to
        ' copy NumElementsToCopy.
        ''''''''''''''''''''''''''''''''''''''
        ReDim ResultArray(1 To DestinationElement + NumElementsToCopy - 1)
    Else
        ''''''''''''''''''''''''''''''''''''''''''''
        ' ResultArray is allocated. If there isn't room
        ' enough in ResultArray to hold NumElementsToCopy
        ' starting at DestinationElement, we need to
        ' resize the array.
        ''''''''''''''''''''''''''''''''''''''''''''
        If (DestinationElement + NumElementsToCopy - 1) > UBound(ResultArray)
Then
            If DestinationElement + NumElementsToCopy > UBound(ResultArray)
Then
                ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                ' Resize the ResultArray.
```

```
            ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
            If NumElementsToCopy + DestinationElement > UBound(ResultArray)
Then
                ReDim Preserve ResultArray(LBound(ResultArray) To
UBound(ResultArray) + DestinationElement - 1)
            Else
                ReDim Preserve ResultArray(LBound(ResultArray) To
UBound(ResultArray) + NumElementsToCopy)
            End If
        Else
            ''''''''''''''''''''''''''''''''''''''''''
            ' Resize the array to hold NumElementsToCopy
            ' starting at DestinationElement.
            ''''''''''''''''''''''''''''''''''''''''''
            ReDim Preserve ResultArray(LBound(ResultArray) To
UBound(ResultArray) + NumElementsToCopy - DestinationElement + 2)
        End If
    Else
        ''''''''''''''''''''''''''''''''''''''''''''''''''''
        ' The ResultArray is large enough to hold
        ' NumberOfElementToCopy starting at DestinationElement.
        ' No need to resize the array.
        ''''''''''''''''''''''''''''''''''''''''''''''''''''''
    End If
  End If
End If


''''''''''''''''''''''''''''''''''''''''''''''''''
' Copy the elements from InputArray to ResultArray
' Note that there is no type compatibility checking
```

```
' when copying the elements.
''''''''''''''''''''''''''''''''''''''''''''''''''

DestNdx = DestinationElement
For SrcNdx = FirstElementToCopy To LastElementToCopy
    If IsObject(InputArray(SrcNdx)) = True Then
        Set ResultArray(DestNdx) = InputArray(DestNdx)
    Else
        On Error Resume Next
        ResultArray(DestNdx) = InputArray(SrcNdx)
        On Error GoTo 0
    End If
    DestNdx = DestNdx + 1
Next SrcNdx

CopyArraySubSetToArray = True

End Function

Public Function NumberOfArrayDimensions(Arr As Variant) As Long
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' NumberOfArrayDimensions
' This function returns the number of dimensions of an array. An unallocated
dynamic array
' has 0 dimensions. This condition can also be tested with IsArrayEmpty.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Dim Ndx As Long
Dim Res As Long
On Error Resume Next
' Loop, increasing the dimension index Ndx, until an error occurs.
' An error will occur when Ndx exceeds the number of dimension
' in the array. Return Ndx - 1.
```

```
Do
    Ndx = Ndx + 1
    Res = UBound(Arr, Ndx)
Loop Until Err.Number <> 0

NumberOfArrayDimensions = Ndx - 1

End Function

Public Function IsArrayDynamic(ByRef Arr As Variant) As Boolean
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' IsArrayDynamic
' This function returns TRUE or FALSE indicating whether Arr is a dynamic array.
' Note that if you attempt to ReDim a static array in the same procedure in which
it is
' declared, you'll get a compiler error and your code won't run at all.
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Dim LUBound As Long

' If we weren't passed an array, get out now with a FALSE result
If IsArray(Arr) = False Then
    IsArrayDynamic = False
    Exit Function
End If

' If the array is empty, it hasn't been allocated yet, so we know
' it must be a dynamic array.
If IsArrayEmpty(Arr:=Arr) = True Then
    IsArrayDynamic = True
    Exit Function
```

```
End If

' Save the UBound of Arr.
' This value will be used to restore the original UBound if Arr
' is a single-dimensional dynamic array. Unused if Arr is multi-dimensional,
' or if Arr is a static array.
LUBound = UBound(Arr)

On Error Resume Next
Err.Clear

' Attempt to increase the UBound of Arr and test the value of Err.Number.
' If Arr is a static array, either single- or multi-dimensional, we'll get a
' C_ERR_ARRAY_IS_FIXED_OR_LOCKED error. In this case, return FALSE.
'
' If Arr is a single-dimensional dynamic array, we'll get C_ERR_NO_ERROR error.
'
' If Arr is a multi-dimensional dynamic array, we'll get a
' C_ERR_SUBSCRIPT_OUT_OF_RANGE error.
'
' For either C_NO_ERROR or C_ERR_SUBSCRIPT_OUT_OF_RANGE, return TRUE.
' For C_ERR_ARRAY_IS_FIXED_OR_LOCKED, return FALSE.

ReDim Preserve Arr(LBound(Arr) To LUBound + 1)

Select Case Err.Number
    Case C_ERR_NO_ERROR
        ' We successfully increased the UBound of Arr.
        ' Do a ReDim Preserve to restore the original UBound.
        ReDim Preserve Arr(LBound(Arr) To LUBound)
        IsArrayDynamic = True
```

```
    Case C_ERR_SUBSCRIPT_OUT_OF_RANGE
        ' Arr is a multi-dimensional dynamic array.
        ' Return True.
        IsArrayDynamic = True
    Case C_ERR_ARRAY_IS_FIXED_OR_LOCKED
        ' Arr is a static single- or multi-dimensional array.
        ' Return False
        IsArrayDynamic = False
    Case Else
        ' We should never get here.
        ' Some unexpected error occurred. Be safe and return False.
        IsArrayDynamic = False
End Select

End Function


Public Function IsArrayEmpty(Arr As Variant) As Boolean
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' IsArrayEmpty
' This function tests whether the array is empty (unallocated). Returns TRUE or
FALSE.
'
' The VBA IsArray function indicates whether a variable is an array, but it does not
' distinguish between allocated and unallocated arrays. It will return TRUE for
both
' allocated and unallocated arrays. This function tests whether the array has
actually
' been allocated.
'
' This function is really the reverse of IsArrayAllocated.
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

Dim LB As Long
Dim UB As Long

Err.Clear
On Error Resume Next
If IsArray(Arr) = False Then
    ' we weren't passed an array, return True
    IsArrayEmpty = True
End If

' Attempt to get the UBound of the array. If the array is
' unallocated, an error will occur.
UB = UBound(Arr, 1)
If (Err.Number <> 0) Then
    IsArrayEmpty = True
Else
    ''''''''''''''''''''''''''''''''''''
    ' On rare occassion, under circumstances I
    ' cannot reliably replictate, Err.Number
    ' will be 0 for an unallocated, empty array.
    ' On these occassions, LBound is 0 and
    ' UBoung is -1.
    ' To accomodate the weird behavior, test to
    ' see if LB > UB. If so, the array is not
    ' allocated.
    ''''''''''''''''''''''''''''''''''''
    Err.Clear
    LB = LBound(Arr)
    If LB > UB Then
```

```
        IsArrayEmpty = True
    Else
        IsArrayEmpty = False
    End If
End If

End Function
```

5. Modules Modules 1

```
Sub Macro1()
'
' Macro1 Macro
'

'
    Selection.NumberFormat = "#,##0.00"
    With Selection
        .HorizontalAlignment = xlGeneral
        .VerticalAlignment = xlBottom
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With
End Sub
```

มหาวิทยาลัยราชพฤกษ์ RAJPHRUK UNIVERSITY

6. Modules 3 Opt

```
Option Explicit

Public Sub Opt3Swap(CustomerArray As Variant, ByVal Bound1 As Long, ByVal
Bound2 As Long, ByVal Bound3 As Long, OptSwapType1_4 As Long)      ' สลับแบบ
3 Opt
    Dim Array1() As Long, Array2() As Long
    Dim Array1_Size As Long, Array2_Size As Long
    Dim i As Long, indexNow As Long

    Array1_Size = Bound2 - Bound1 + 1: Array2_Size = Bound3 - Bound2
    ReDim Array1(1 To Array1_Size): ReDim Array2(1 To Array2_Size)    'เตรียม
ขนาด Array เท่ากับช่วงข้อมูลที่ 1 และ 2

    indexNow = 1    'เก็บข้อมูล Array ช่วง 1
    For i = Bound1 To Bound2
        Array1(indexNow) = CustomerArray(i):  indexNow = indexNow + 1
    Next i

    indexNow = 1    'เก็บข้อมูล Array ช่วง 2
    For i = Bound2 + 1 To Bound3
        Array2(indexNow) = CustomerArray(i):  indexNow = indexNow + 1
    Next i

    If (OptSwapType1_4 = 1) Then
        Call SwapBit(Array1, 1, Array1_Size)
        indexNow = Bound1    'เขียนค่าข้อมูล Array ช่วง 1
        For i = 1 To Array1_Size
```

```
            CustomerArray(indexNow) = Array1(i):  indexNow = indexNow + 1
        Next i

        Call SwapBit(Array2, 1, Array2_Size)
        'indexNow = Bound2    'เขียนค่าข้อมูล Array ช่วง 2
        For i = 1 To Array2_Size
            CustomerArray(indexNow) = Array2(i):  indexNow = indexNow + 1
        Next i

    ElseIf (OptSwapType1_4 = 2) Then
        indexNow = Bound1    'เขียนค่าข้อมูล Array ช่วง 1
        For i = 1 To Array2_Size
            CustomerArray(indexNow) = Array2(i):  indexNow = indexNow + 1
        Next i

        Call SwapBit(Array1, 1, Array1_Size)
        'indexNow = Bound2    'เขียนค่าข้อมูล Array ช่วง 2
        For i = 1 To Array1_Size
            CustomerArray(indexNow) = Array1(i):  indexNow = indexNow + 1
        Next i
    ElseIf (OptSwapType1_4 = 3) Then
        Call SwapBit(Array2, 1, Array2_Size)
        indexNow = Bound1    'เขียนค่าข้อมูล Array ช่วง 1
        For i = 1 To Array2_Size
            CustomerArray(indexNow) = Array2(i):  indexNow = indexNow + 1
        Next i

        'indexNow = Bound2    'เขียนค่าข้อมูล Array ช่วง 2
        For i = 1 To Array1_Size
            CustomerArray(indexNow) = Array1(i):  indexNow = indexNow + 1
        Next i
```

```
        Else
            indexNow = Bound1    'เขียนค่าข้อมูล Array ช่วง 1
            For i = 1 To Array2_Size
                CustomerArray(indexNow) = Array2(i):  indexNow = indexNow + 1
            Next i

            'indexNow = Bound2    'เขียนค่าข้อมูล Array ช่วง 2
            For i = 1 To Array1_Size
                CustomerArray(indexNow) = Array1(i):  indexNow = indexNow + 1
            Next i
        End If

End Sub


Private Function Cal3Opt(ResultArray As Variant) As Boolean        'คืนค่าการปรับปรุง
คำตอบทั้งหมด เป็น True ถ้า ปรับปรุงให้ดีขึ้น
        Dim StationQuant As Long
        Dim ArrayBestResult() As Long, arrayTrySwapOpt() As Long
        Dim DistanceOrigin As Double, DistanceBest As Double, DistanceOpt As
Double

        Dim FoundBetterResult As Boolean
        Dim StartIndex As Long, EndIndex As Long

        Dim i As Long, j As Long, k As Long

        Dim RoundCal As Long

        Dim AllBetterResult As Boolean   'การปรับปรุงคำตอบทั้งหมดว่าดีขึ้นหรือไม่
```

```
        AllBetterResult = False

        StationQuant = UBound(ResultArray)    'จำนวนลูกค้า
        ReDim ArrayBestResult(1 To StationQuant)     'เตรียมขนาด Array เท่ากับจำนวน
ลูกค้า
        ReDim arrayTrySwapOpt(1 To StationQuant)

        Call Array_Copy(ResultArray, ArrayBestResult)     'คัดลอกลำดับการจัดส่ง  ถือว่า
ตัวเริ่มต้นเป็นตัวที่ดีที่สุด
        DistanceOrigin = Tour_Distance(ArrayBestResult) 'คำนวณระยะทางจาก ลำดับการ
จัดส่ง
        DistanceBest = DistanceOrigin

        FoundBetterResult = True
        RoundCal = 0
        Do Until (FoundBetterResult = False)    'ทำจนกว่าไม่สามารถ ปรับปรุงค่าคำตอบได้
            FoundBetterResult = False  'คิดว่าไม่เจอคำตอบที่ดีกว่าไว้ก่อน
            RoundCal = RoundCal + 1
            'ActiveWorkbook.Worksheets("Data").Range("I13").value = RoundCal


            For i = 1 To StationQuant - 1
                For j = i To StationQuant - 1
                    For k = j + 1 To StationQuant
                        '================  3-OPT  แบบที่ 1 Start
================
                        Call Array_Copy(ResultArray, arrayTrySwapOpt)
'คัดลอกลูกค้าจากตัวเริ่มต้นใน Itteration ไปยังตัวที่จะทำ  Opt
                        Call Opt3Swap(arrayTrySwapOpt, i, j, k, 1)  'ทำ 3
Opt
```

```
                                DistanceOpt = Tour_Distance(arrayTrySwapOpt)
'คำนวณระยะทางจาก Trip

                                If (DistanceOpt < DistanceBest) Then     'ถ้าใช้ 2
Opt แล้วค่าดีขึ้น เก็บแผนการขนส่งดังกล่าว
                                    DistanceBest = DistanceOpt
                                    Call Array_Copy(arrayTrySwapOpt,
ArrayBestResult)
                                    FoundBetterResult = True     'แสดงว่ารอบนี้
ปรับปรุงค่าคำตอบได้
                                End If
                                '================  3-OPT  แบบที่ 1 End
================

                                '================  3-OPT  แบบที่ 2 Start
================
                                Call Array_Copy(ResultArray, arrayTrySwapOpt)
'คัดลอกลูกค้าจากตัวเริ่มต้นใน Itteration ไปยังตัวที่จะทำ  Opt
                                Call Opt3Swap(arrayTrySwapOpt, i, j, k, 2)  'ทำ 3
Opt
                                DistanceOpt = Tour_Distance(arrayTrySwapOpt)
'คำนวณระยะทางจาก Trip

                                If (DistanceOpt < DistanceBest) Then     'ถ้าใช้ 2
Opt แล้วค่าดีขึ้น เก็บแผนการขนส่งดังกล่าว
                                    DistanceBest = DistanceOpt
                                    Call Array_Copy(arrayTrySwapOpt,
ArrayBestResult)
                                    FoundBetterResult = True     'แสดงว่ารอบนี้
ปรับปรุงค่าคำตอบได้
                                End If
```

```
                    '================  3-OPT  แบบที่ 2 End
================

                    '================  3-OPT  แบบที่ 3 Start
================
                    Call Array_Copy(ResultArray, arrayTrySwapOpt)
'คัดลอกลูกค้าจากตัวเริ่มต้นใน Itteration ไปยังตัวที่จะทำ  Opt
                    Call Opt3Swap(arrayTrySwapOpt, i, j, k, 3)  'ทำ 3
Opt
                    DistanceOpt = Tour_Distance(arrayTrySwapOpt)
'คำนวณระยะทางจาก Trip

                    If (DistanceOpt < DistanceBest) Then      'ถ้าใช้ 2
Opt แล้วค่าดีขึ้น เก็บแผนการขนส่งดังกล่าว
                        DistanceBest = DistanceOpt
                        Call Array_Copy(arrayTrySwapOpt,
ArrayBestResult)
                        FoundBetterResult = True     'แสดงว่ารอบนี้
ปรับปรุงค่าคำตอบได้
                    End If
                    '================  3-OPT  แบบที่ 3 End
================

                    '================  3-OPT  แบบที่ 4 Start
================
                    Call Array_Copy(ResultArray, arrayTrySwapOpt)
'คัดลอกลูกค้าจากตัวเริ่มต้นใน Itteration ไปยังตัวที่จะทำ  Opt
                    Call Opt3Swap(arrayTrySwapOpt, i, j, k, 4)  'ทำ 3
Opt
                    DistanceOpt = Tour_Distance(arrayTrySwapOpt)
'คำนวณระยะทางจาก Trip
```

```
                                If (DistanceOpt < DistanceBest) Then      'ถ้าใช้ 2
Opt แล้วค่าดีขึ้น เก็บแผนการขนส่งดังกล่าว
                                    DistanceBest = DistanceOpt
                                    Call Array_Copy(arrayTrySwapOpt,
ArrayBestResult)
                                    FoundBetterResult = True     'แสดงว่ารอบนี้
ปรับปรุงค่าคำตอบได้
                                End If
                                '================  3-OPT  แบบที่ 4 End
================

                            Next k
                        Next j
                Next i

                If (FoundBetterResult) Then
                        Call Array_Copy(ArrayBestResult, ResultArray)
                        AllBetterResult = True   'ค่าการปรับปรุงคำตอบทั้งหมด True
เพราะ ปรับปรุงให้ดีขึ้น
                End If
        Loop
        Cal3Opt = AllBetterResult   'คืนค่าการปรับปรุงคำตอบทั้งหมด True ถ้า ปรับปรุงให้
ดีขึ้น
End Function


Public Sub Cal3OptLoad()      'Load แบบ 3Opt
    Dim CarQty As Long, RowQty As Long
    Dim iCar As Long
    Dim aCustomer() As Long
```

```
    Dim rNearLess As Range, rOpt As Range
    Dim CarModifySolution As Long
    Dim aStation() As Long   'ลูกค้าที่จัดส่ง
    Dim FoundBetterResult As Boolean

    Dim startRow As Long, endRow As Long

    wsResult.Range("BC11:BL10000").Delete Shift:=xlUp  'ลบข้อมูลเดิม
    CarQty = wsResult.Range("B19").value
    RowQty = wsResult.Range("C19").value

    Set rNearLess = wsResult.Range("AR10")
    Set rOpt = wsResult.Range("BD10")

    Dim i As Long

    startRow = 11
    For CarModifySolution = 1 To CarQty
        endRow = ReadStationInCar("AX", "AR", CarModifySolution, aStation)

        'rOrderCarLoadProduct.Offset(RowWriteUseNow - 10, 6).value =
Tour_Distance(aStation)
        FoundBetterResult = True
        Do Until (FoundBetterResult = False)    'ทำจนกว่าไม่สามารถ ปรับปรุงค่าคำตอบได้
                FoundBetterResult = False
                FoundBetterResult = Cal3Opt(aStation)
        Loop


        wsResult.Range("AR" & CStr(startRow) & ":AZ" & CStr(endRow)).Copy
Destination:=wsResult.Range("BD" & CStr(startRow))
```

```
        wsResult.Range("BK" & CStr(endRow)).value = Tour_Distance(aStation)
        wsResult.Range("BK" & CStr(endRow)).NumberFormat = "#,##0.00"

        For i = startRow To endRow
            wsResult.Range("BC" & CStr(i)).value = FindStationInArray(aStation,
wsResult.Range("BD" & CStr(i)).value)
        Next i


       ' เรียงจาก Customer ID แล้วตามด้วย ปริมาตร มาก ไป น้อย
        ActiveWorkbook.Worksheets("Result").Sort.SortFields.Clear
        ActiveWorkbook.Worksheets("Result").Sort.SortFields.Add Key:=Range( _
                "BC" & CStr(startRow) & ":BC" & CStr(endRow)), SortOn:=xlSortOnValues,
Order:=xlAscending, DataOption:=xlSortNormal   '"Worksheet Result" กำหนดให้
ลำดับการส่ง Column BC เป็น Index
        With ActiveWorkbook.Worksheets("Result").Sort   ' เรียงข้อมูลสินค้าทั้งหมด ตั้งแต่
แถวที่ 11
                .SetRange Range("BC" & CStr(startRow) & ":BJ" & CStr(endRow))
                .Header = xlGuess
                .MatchCase = False
                .Orientation = xlTopToBottom
                .SortMethod = xlPinYin
                .Apply
        End With

       startRow = endRow + 1
     Next CarModifySolution
    End Sub
```

7. Module SA

```
Private Function xxxCalSA2(ResultArray As Variant, tempStart As Double,
coolingRate As Double, boltzmannK As Double) As Boolean
    Dim StationQuant As Long
    Dim ArrayChallengeResult() As Long
    Dim DistanceOrigin As Double, DistanceChallenge As Double
    Dim position1 As Long, position2 As Long, position3 As Long
    Dim tempCity As Long
    Dim tempNow As Double  ' อุณหภูมิ ปัจจุบัน
    Dim randomVal As Double, acceptVal As Double

    StationQuant = UBound(ResultArray)    'จำนวนลูกค้า
    ReDim ArrayChallengeResult(1 To StationQuant)     'เตรียมขนาด Array เท่ากับ
จำนวนลูกค้า
    DistanceOrigin = Tour_Distance(ResultArray) 'คำนวณระยะทางจาก ลำดับการจัดส่ง

    tempNow = tempStart

   Do While tempNow >= 1#
        Call Array_Copy(ResultArray, ArrayChallengeResult)
        position1 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 1
        position2 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 2

        tempCity = ArrayChallengeResult(position1)
        ArrayChallengeResult(position1) = ArrayChallengeResult(position2)
        ArrayChallengeResult(position2) = tempCity
        DistanceChallenge = Tour_Distance(ArrayChallengeResult) '

        If DistanceChallenge < DistanceOrigin Then
            DistanceOrigin = DistanceChallenge
            Call Array_Copy(ArrayChallengeResult, ResultArray)
        Else
```

```
            randomVal = Rnd()
            acceptVal = Exp((DistanceOrigin - DistanceChallenge) / (tempNow *
boltzmannK))
            If randomVal < acceptVal Then
                DistanceOrigin = DistanceChallenge
                Call Array_Copy(ArrayChallengeResult, ResultArray)
            End If
        End If

        tempNow = (1 - coolingRate) * tempNow
    Loop

End Function
Private Function xxxCalSA3(ResultArray As Variant, tempStart As Double,
coolingRate As Double, boltzmannK As Double) As Boolean
    Dim StationQuant As Long
    Dim ArrayChallengeResult() As Long
    Dim DistanceOrigin As Double, DistanceChallenge As Double
    Dim position1 As Long, position2 As Long, position3 As Long
    Dim tempCity As Long
    Dim tempNow As Double  ' อุณหภูมิ ปัจจุบัน
    Dim randomVal As Double, acceptVal As Double

    StationQuant = UBound(ResultArray)    'จำนวนลูกค้า
    ReDim ArrayChallengeResult(1 To StationQuant)     'เตรียมขนาด Array เท่ากับ
จำนวนลูกค้า
    DistanceOrigin = Tour_Distance(ResultArray) 'คำนวณระยะทางจาก ลำดับการจัดส่ง

    tempNow = tempStart

   Do While tempNow >= 1#
```

```
        Call Array_Copy(ResultArray, ArrayChallengeResult)
        position1 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 1
        position2 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 2
        position3 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 3

        tempCity = ArrayChallengeResult(position1)
        ArrayChallengeResult(position1) = ArrayChallengeResult(position2)
        ArrayChallengeResult(position2) = ArrayChallengeResult(position3)
        ArrayChallengeResult(position3) = tempCity
        DistanceChallenge = Tour_Distance(ArrayChallengeResult) '

        If DistanceChallenge < DistanceOrigin Then
            DistanceOrigin = DistanceChallenge
            Call Array_Copy(ArrayChallengeResult, ResultArray)
        Else
            randomVal = Rnd()
            acceptVal = Exp((DistanceOrigin - DistanceChallenge) / (tempNow *
boltzmannK))
            If randomVal < acceptVal Then
                DistanceOrigin = DistanceChallenge
                Call Array_Copy(ArrayChallengeResult, ResultArray)
            End If
        End If

        tempNow = (1 - coolingRate) * tempNow
    Loop

End Function

Private Function xxxCalSA4(ResultArray As Variant, tempStart As Double,
coolingRate As Double, boltzmannK As Double) As Boolean
```

```
    Dim StationQuant As Long
    Dim ArrayChallengeResult() As Long
    Dim DistanceOrigin As Double, DistanceChallenge As Double
    Dim position1 As Long, position2 As Long, position3 As Long, position4 As Long
    Dim tempCity As Long
    Dim tempNow As Double  ' อุณหภูมิ ปัจจุบัน
    Dim randomVal As Double, acceptVal As Double
        StationQuant = UBound(ResultArray)    'จำนวนลูกค้า
    ReDim ArrayChallengeResult(1 To StationQuant)     'เตรียมขนาด Array เท่ากับ
จำนวนลูกค้า
    DistanceOrigin = Tour_Distance(ResultArray) 'คำนวณระยะทางจาก ลำดับการจัดส่ง
    tempNow = tempStart
   Do While tempNow >= 1#
        Call Array_Copy(ResultArray, ArrayChallengeResult)
        position1 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 1
        position2 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 2
        position3 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 3
        position4 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 4
        tempCity = ArrayChallengeResult(position1)
        ArrayChallengeResult(position1) = ArrayChallengeResult(position2)
        ArrayChallengeResult(position2) = ArrayChallengeResult(position3)
        ArrayChallengeResult(position3) = ArrayChallengeResult(position4)
        ArrayChallengeResult(position4) = tempCity
        DistanceChallenge = Tour_Distance(ArrayChallengeResult) '
        If DistanceChallenge < DistanceOrigin Then
            DistanceOrigin = DistanceChallenge
            Call Array_Copy(ArrayChallengeResult, ResultArray)
        Else
            randomVal = Rnd()
            acceptVal = Exp((DistanceOrigin - DistanceChallenge) / (tempNow *
boltzmannK))
```

```
            If randomVal < acceptVal Then
                DistanceOrigin = DistanceChallenge
                Call Array_Copy(ArrayChallengeResult, ResultArray)
            End If
        End If
        tempNow = (1 - coolingRate) * tempNow
    Loop

End Function
Private Function xxxCalSA5(ResultArray As Variant, tempStart As Double,
coolingRate As Double, boltzmannK As Double) As Boolean
    Dim StationQuant As Long
    Dim ArrayChallengeResult() As Long
    Dim DistanceOrigin As Double, DistanceChallenge As Double
    Dim position1 As Long, position2 As Long, position3 As Long, position4 As
Long, position5 As Long
    Dim tempCity As Long
    Dim tempNow As Double  ' อุณหภูมิ ปัจจุบัน
    Dim randomVal As Double, acceptVal As Double
    StationQuant = UBound(ResultArray)    'จำนวนลูกค้า
    ReDim ArrayChallengeResult(1 To StationQuant)     'เตรียมขนาด Array เท่ากับ
จำนวนลูกค้า
    DistanceOrigin = Tour_Distance(ResultArray) 'คำนวณระยะทางจาก ลำดับการจัดส่ง

    tempNow = tempStart
   Do While tempNow >= 1#
        Call Array_Copy(ResultArray, ArrayChallengeResult)
        position1 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 1
        position2 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 2
        position3 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 3
        position4 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 4
```

```
        position5 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 4
        tempCity = ArrayChallengeResult(position1)
        ArrayChallengeResult(position1) = ArrayChallengeResult(position2)
        ArrayChallengeResult(position2) = ArrayChallengeResult(position3)
        ArrayChallengeResult(position3) = ArrayChallengeResult(position4)
        ArrayChallengeResult(position4) = ArrayChallengeResult(position5)
        ArrayChallengeResult(position5) = tempCity
        DistanceChallenge = Tour_Distance(ArrayChallengeResult) '
        If DistanceChallenge < DistanceOrigin Then
            DistanceOrigin = DistanceChallenge
            Call Array_Copy(ArrayChallengeResult, ResultArray)
        Else
            randomVal = Rnd()
            acceptVal = Exp((DistanceOrigin - DistanceChallenge) / (tempNow *
boltzmannK))
            If randomVal < acceptVal Then
                DistanceOrigin = DistanceChallenge
                Call Array_Copy(ArrayChallengeResult, ResultArray)
            End If
        End If
tempNow = (1 - coolingRate) * tempNow
    Loop
End Function
Public Sub xxxCalSALoad()      'Load แบบ SA
    Dim CarQty As Long, RowQty As Long
    Dim iCar As Long
    Dim aCustomer() As Long
    Dim rNearLess As Range, rSA As Range
    Dim CarModifySolution As Long
    Dim aStation() As Long   'ลูกค้าที่จัดส่ง
    Dim FoundBetterResult As Boolean
```

```
    Dim startRow As Long, endRow As Long
    Dim tempStart As Double, coolingRate As Double
    Dim swapPoint As String
    Dim boltzmannK As Double
    wsResult.Range("BO11:BX10000").Delete Shift:=xlUp  'ลบข้อมูลเดิม
    CarQty = wsResult.Range("B19").value
    RowQty = wsResult.Range("C19").value
    Set rNearLess = wsResult.Range("AR10")
    Set rSA = wsResult.Range("BP10")
    Dim i As Long
    swapPoint = wsCalMain.Range("O2").value
    tempStart = wsCalMain.Range("O3").value
    coolingRate = wsCalMain.Range("P3").value
    boltzmannK = wsCalMain.Range("O4").value
      startRow = 11
    For CarModifySolution = 1 To CarQty
        endRow = ReadStationInCar("AX", "AR", CarModifySolution, aStation)
        If swapPoint = "2 Point" Then
            Call CalSA2(aStation, tempStart, coolingRate, boltzmannK)
        ElseIf swapPoint = "3 Point" Then
            Call CalSA3(aStation, tempStart, coolingRate, boltzmannK)
        ElseIf swapPoint = "4 Point" Then
            Call CalSA4(aStation, tempStart, coolingRate, boltzmannK)
        ElseIf swapPoint = "5 Point" Then
            Call CalSA5(aStation, tempStart, coolingRate, boltzmannK)
        End If
        wsResult.Range("AR" & CStr(startRow) & ":AZ" & CStr(endRow)).Copy
Destination:=wsResult.Range("BP" & CStr(startRow))
        wsResult.Range("BW" & CStr(endRow)).value = Tour_Distance(aStation)
        wsResult.Range("BW" & CStr(endRow)).NumberFormat = "#,##0.00"
        For i = startRow To endRow
```

```
            wsResult.Range("BO" & CStr(i)).value = FindStationInArray(aStation,
wsResult.Range("BP" & CStr(i)).value)
        Next i
            ActiveWorkbook.Worksheets("Result").Sort.SortFields.Clear
        ActiveWorkbook.Worksheets("Result").Sort.SortFields.Add Key:=Range( _
            "BO" & CStr(startRow) & ":BO" & CStr(endRow)),
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
'"Worksheet Result" กำหนดให้ ลำดับการส่ง Column BO เป็น Index
        With ActiveWorkbook.Worksheets("Result").Sort  ' เรียงข้อมูลสินค้าทั้งหมด ตั้งแต่
แถวที่ 11
            .SetRange Range("BO" & CStr(startRow) & ":BV" & CStr(endRow))
            .Header = xlGuess
            .MatchCase = False
            .Orientation = xlTopToBottom
            .SortMethod = xlPinYin
            .Apply
        End With

       startRow = endRow + 1
    Next CarModifySolution



End Sub



Private Function CalSA2(ResultArray As Variant, tempStart As Double, coolingRate
As Double, boltzmannK As Double) As Boolean
    Dim StationQuant As Long
    Dim ArrayChallengeResult() As Long
```

```
    Dim DistanceOrigin As Double, DistanceChallenge As Double
    Dim position1 As Long, position2 As Long, position3 As Long
    Dim tempCity As Long
    Dim tempNow As Double  ' อุณหภูมิ ปัจจุบัน
    Dim randomVal As Double, acceptVal As Double
    StationQuant = UBound(ResultArray)    'จำนวนลูกค้า
    ReDim ArrayChallengeResult(1 To StationQuant)     'เตรียมขนาด Array เท่ากับ
จำนวนลูกค้า
    DistanceOrigin = Tour_Distance(ResultArray) 'คำนวณระยะทางจาก ลำดับการจัดส่ง
    tempNow = tempStart
   Do While tempNow >= 1#
        Call Array_Copy(ResultArray, ArrayChallengeResult)
        position1 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 1
        position2 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 2
        tempCity = ArrayChallengeResult(position1)
        ArrayChallengeResult(position1) = ArrayChallengeResult(position2)
        ArrayChallengeResult(position2) = tempCity
        DistanceChallenge = Tour_Distance(ArrayChallengeResult) '
        If DistanceChallenge < DistanceOrigin Then
            DistanceOrigin = DistanceChallenge
            Call Array_Copy(ArrayChallengeResult, ResultArray)
        Else
            randomVal = Rnd()
            acceptVal = Exp((DistanceOrigin - DistanceChallenge) / (tempNow *
boltzmannK))
            If randomVal < acceptVal Then
                DistanceOrigin = DistanceChallenge
                Call Array_Copy(ArrayChallengeResult, ResultArray)
            End If
        End If
```

```
        tempNow = (1 - coolingRate) * tempNow
    Loop
End Function
Private Function CalSA3(ResultArray As Variant, tempStart As Double, coolingRate
As Double, boltzmannK As Double) As Boolean
    Dim StationQuant As Long
    Dim ArrayChallengeResult() As Long
    Dim DistanceOrigin As Double, DistanceChallenge As Double
    Dim position1 As Long, position2 As Long, position3 As Long
    Dim tempCity As Long
    Dim tempNow As Double  ' อุณหภูมิ ปัจจุบัน
    Dim randomVal As Double, acceptVal As Double
    StationQuant = UBound(ResultArray)    'จำนวนลูกค้า
    ReDim ArrayChallengeResult(1 To StationQuant)     'เตรียมขนาด Array เท่ากับ
จำนวนลูกค้า
    DistanceOrigin = Tour_Distance(ResultArray) 'คำนวณระยะทางจาก ลำดับการจัดส่ง
    tempNow = tempStart
   Do While tempNow >= 1#
        Call Array_Copy(ResultArray, ArrayChallengeResult)
        position1 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 1
        position2 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 2
        position3 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 3
        tempCity = ArrayChallengeResult(position1)
        ArrayChallengeResult(position1) = ArrayChallengeResult(position2)
        ArrayChallengeResult(position2) = ArrayChallengeResult(position3)
        ArrayChallengeResult(position3) = tempCity
        DistanceChallenge = Tour_Distance(ArrayChallengeResult) '
        If DistanceChallenge < DistanceOrigin Then
            DistanceOrigin = DistanceChallenge
            Call Array_Copy(ArrayChallengeResult, ResultArray)
        Else
```

```
            randomVal = Rnd()
            acceptVal = Exp((DistanceOrigin - DistanceChallenge) / (tempNow *
boltzmannK))
            If randomVal < acceptVal Then
                DistanceOrigin = DistanceChallenge
                Call Array_Copy(ArrayChallengeResult, ResultArray)
            End If
        End If
            tempNow = (1 - coolingRate) * tempNow
    Loop
End Function
Private Function CalSA4(ResultArray As Variant, tempStart As Double, coolingRate
As Double, boltzmannK As Double) As Boolean
    Dim StationQuant As Long
    Dim ArrayChallengeResult() As Long
    Dim DistanceOrigin As Double, DistanceChallenge As Double
    Dim position1 As Long, position2 As Long, position3 As Long, position4 As Long
    Dim tempCity As Long
    Dim tempNow As Double  ' อุณหภูมิ ปัจจุบัน
    Dim randomVal As Double, acceptVal As Double
    StationQuant = UBound(ResultArray)    'จำนวนลูกค้า
    ReDim ArrayChallengeResult(1 To StationQuant)     'เตรียมขนาด Array เท่ากับ
จำนวนลูกค้า
    DistanceOrigin = Tour_Distance(ResultArray) 'คำนวณระยะทางจาก ลำดับการจัดส่ง
    tempNow = tempStart
   Do While tempNow >= 1#
        Call Array_Copy(ResultArray, ArrayChallengeResult)
        position1 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 1
        position2 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 2
        position3 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 3
        position4 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 4
```

```
        tempCity = ArrayChallengeResult(position1)
        ArrayChallengeResult(position1) = ArrayChallengeResult(position2)
        ArrayChallengeResult(position2) = ArrayChallengeResult(position3)
        ArrayChallengeResult(position3) = ArrayChallengeResult(position4)
        ArrayChallengeResult(position4) = tempCity
        DistanceChallenge = Tour_Distance(ArrayChallengeResult) '
        If DistanceChallenge < DistanceOrigin Then
            DistanceOrigin = DistanceChallenge
            Call Array_Copy(ArrayChallengeResult, ResultArray)
        Else
            randomVal = Rnd()
            acceptVal = Exp((DistanceOrigin - DistanceChallenge) / (tempNow *
boltzmannK))
            If randomVal < acceptVal Then
                DistanceOrigin = DistanceChallenge
                Call Array_Copy(ArrayChallengeResult, ResultArray)
            End If
        End If
        tempNow = (1 - coolingRate) * tempNow
    Loop
End Function
Private Function CalSA5(ResultArray As Variant, tempStart As Double, coolingRate
As Double, boltzmannK As Double) As Boolean
    Dim StationQuant As Long
    Dim ArrayChallengeResult() As Long
    Dim DistanceOrigin As Double, DistanceChallenge As Double
    Dim position1 As Long, position2 As Long, position3 As Long, position4 As
Long, position5 As Long
    Dim tempCity As Long
    Dim tempNow As Double  ' อุณหภูมิ ปัจจุบัน
```

```
    Dim randomVal As Double, acceptVal As Doubl
StationQuant = UBound(ResultArray)    'จำนวนลูกค้า
ReDim ArrayChallengeResult(1 To StationQuant)     'เตรียมขนาด Array เท่ากับจำนวน
ลูกค้า
    DistanceOrigin = Tour_Distance(ResultArray) 'คำนวณระยะทางจาก ลำดับการจัดส่ง
    tempNow = tempStart
   Do While tempNow >= 1#
        Call Array_Copy(ResultArray, ArrayChallengeResult)
        position1 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 1
        position2 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 2
        position3 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 3
        position4 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 4
        position5 = CLng(Int((StationQuant) * Rnd() + 1)) 'สุ่มจุดตัดที่ 4
        tempCity = ArrayChallengeResult(position1)
        ArrayChallengeResult(position1) = ArrayChallengeResult(position2)
        ArrayChallengeResult(position2) = ArrayChallengeResult(position3)
        ArrayChallengeResult(position3) = ArrayChallengeResult(position4)
        ArrayChallengeResult(position4) = ArrayChallengeResult(position5)
        ArrayChallengeResult(position5) = tempCity
        DistanceChallenge = Tour_Distance(ArrayChallengeResult) '
            If DistanceChallenge < DistanceOrigin Then
           DistanceOrigin = DistanceChallenge
           Call Array_Copy(ArrayChallengeResult, ResultArray)
        Else
           randomVal = Rnd()
           acceptVal = Exp((DistanceOrigin - DistanceChallenge) / (tempNow *
boltzmannK))
           If randomVal < acceptVal Then
              DistanceOrigin = DistanceChallenge
              Call Array_Copy(ArrayChallengeResult, ResultArray)
           End If
```

```
        End If

        tempNow = (1 - coolingRate) * tempNow
    Loop
End Function
Public Sub CalSALoad()      'Load แบบ SA
    Dim CarQty As Long, RowQty As Long
    Dim iCar As Long
    Dim aCustomer() As Long
    Dim rNearLess As Range, rSA As Range
    Dim CarModifySolution As Long
    Dim aStation() As Long   'ลูกค้าที่จัดส่ง
    Dim FoundBetterResult As Boolean
    Dim startRow As Long, endRow As Long
    Dim tempStart As Double, coolingRate As Double
    Dim swapPoint As String
    Dim boltzmannK As Double
    wsResult.Range("BO11:BX10000").Delete Shift:=xlUp  'ลบข้อมูลเดิม
    CarQty = wsResult.Range("B19").value
    RowQty = wsResult.Range("C19").value
    Set rNearLess = wsResult.Range("AR10")
    Set rSA = wsResult.Range("BP10")
    Dim i As Long
    swapPoint = wsCalMain.Range("O2").value
    tempStart = wsCalMain.Range("O3").value
    coolingRate = wsCalMain.Range("P3").value
    boltzmannK = wsCalMain.Range("O4").value
    'xxxxxxxxx 2
    startRow = 11
    For CarModifySolution = 1 To CarQty
        endRow = ReadStationInCar("AX", "AR", CarModifySolution, aStation)
```

```
Call CalSA2(aStation, tempStart, coolingRate, boltzmannK)
wsResult.Range("AR" & CStr(startRow) & ":AZ" & CStr(endRow)).Copy
Destination:=wsResult.Range("BP" & CStr(startRow))
wsResult.Range("BW" & CStr(endRow)).value = Tour_Distance(aStation)
wsResult.Range("BW" & CStr(endRow)).NumberFormat = "#,##0.00"
For i = startRow To endRow
wsResult.Range("BO" & CStr(i)).value = FindStationInArray(aStation,
wsResult.Range("BP" & CStr(i)).value) Next i
        ActiveWorkbook.Worksheets("Result").Sort.SortFields.Clear
        ActiveWorkbook.Worksheets("Result").Sort.SortFields.Add Key:=Range( _
"BO" & CStr(startRow) & ":BO" & CStr(endRow)), SortOn:=xlSortOnValues,
Order:=xlAscending, DataOption:=xlSortNormal  '"Worksheet Result" กำหนดให้
ลำดับการส่ง Column BO เป็น Index With ActiveWorkbook.Worksheets("Result").Sort
' เรียงข้อมูลสินค้าทั้งหมด ตั้งแต่ แถวที่ 11
.SetRange Range("BO" & CStr(startRow) & ":BV" & CStr(endRow))
.Header = xlGuess.MatchCase = False
.Orientation = xlTopToBottom.SortMethod = xlPinYin
.ApplyEnd WithstartRow = endRow + 1
Next CarModifySolutionwsResult.Range("BW8").Formula = "=SUM(BW11:BW10000)":
wsResult.Range("BW8").value = wsResult.Range("BW8").value
wsResult.Range("BW8").NumberFormat = "#,##0.00"'xxxxxxxxx 3
 startRow = 11For CarModifySolution = 1 To CarQty
endRow = ReadStationInCar("AX", "AR", CarModifySolution, aStation)
Call CalSA3(aStation, tempStart, coolingRate, boltzmannK)
wsResult.Range("AR" & CStr(startRow) & ":AZ" & CStr(endRow)).Copy
Destination:=wsResult.Range("BP" & CStr(startRow))
wsResult.Range("BW" & CStr(endRow)).value = Tour_Distance(aStation)
wsResult.Range("BW" & CStr(endRow)).NumberFormat = "#,##0.00"
For i = startRow To endRow
```

```
wsResult.Range("BO" & CStr(i)).value = FindStationInArray(aStation, wsResult.Range("BP"
& CStr(i)).value) Next I ActiveWorkbook.Worksheets("Result").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("Result").Sort.SortFields.Add Key:=Range( _
"BO" & CStr(startRow) & ":BO" & CStr(endRow)), SortOn:=xlSortOnValues,
Order:=xlAscending, DataOption:=xlSortNormal  '"Worksheet Result" กำหนดให้ ลำดับการ
ส่ง Column BO เป็น Index With ActiveWorkbook.Worksheets("Result").Sort  ' เรียงข้อมูล
สินค้าทั้งหมด ตั้งแต่ แถวที่ 11.SetRange Range("BO" & CStr(startRow) & ":BV" &
CStr(endRow))
.Header = xlGuess.MatchCase = False
.Orientation = xlTopToBottom .SortMethod = xlPinYin
Apply End With startRow = endRow + 1
Next CarModifySolution
wsResult.Range("BW8").Formula = "=SUM(BW11:BW10000)":
wsResult.Range("BW8").value = wsResult.Range("BW8").value
wsResult.Range("BW8").NumberFormat = "#,##0.00" 'xxxxxxxxx 4
startRow = 11 For CarModifySolution = 1 To CarQty
endRow = ReadStationInCar("AX", "AR", CarModifySolution, aStation)
Call CalSA4(aStation, tempStart, coolingRate, boltzmannK)
wsResult.Range("AR" & CStr(startRow) & ":AZ" & CStr(endRow)).Copy
Destination:=wsResult.Range("BP" & CStr(startRow))
wsResult.Range("BW" & CStr(endRow)).value = Tour_Distance(aStation)
wsResult.Range("BW" & CStr(endRow)).NumberFormat = "#,##0.00"
For i = startRow To endRow
wsResult.Range("BO" & CStr(i)).value = FindStationInArray(aStation, wsResult.Range("BP"
& CStr(i)).value)Next i
ActiveWorkbook.Worksheets("Result").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("Result").Sort.SortFields.Add Key:=Range( _
"BO" & CStr(startRow) & ":BO" & CStr(endRow)), SortOn:=xlSortOnValues,
Order:=xlAscending, DataOption:=xlSortNormal  '"Worksheet Result" กำหนดให้ ลำดับการ
ส่ง Column BO เป็น Index
With ActiveWorkbook.Worksheets("Result").Sort  ' เรียงข้อมูลสินค้าทั้งหมด ตั้งแต่ แถวที่ 11
```

```
SetRange Range("BO" & CStr(startRow) & ":BV" & CStr(endRow))
.Header = xlGuess.MatchCase = False.Orientation = xlTopToBottom.SortMethod =
xlPinYin.ApplyEnd With
    startRow = endRow + 1Next CarModifySolution
    wsResult.Range("BW8").Formula = "=SUM(BW11:BW10000)":
    wsResult.Range("BW8").value = wsResult.Range("BW8").value
    wsResult.Range("BW8").NumberFormat = "#,##0.00"
    'xxxxxxxxx 5 startRow = 11
    For CarModifySolution = 1 To CarQty
    endRow = ReadStationInCar("AX", "AR", CarModifySolution, aStation)
    Call CalSA5(aStation, tempStart, coolingRate, boltzmannK)
    wsResult.Range("AR" & CStr(startRow) & ":AZ" & CStr(endRow)).Copy
    Destination:=wsResult.Range("BP" & CStr(startRow))
    wsResult.Range("BW" & CStr(endRow)).value = Tour_Distance(aStation)
    wsResult.Range("BW" & CStr(endRow)).NumberFormat = "#,##0.00"
    For i = startRow To endRow
    wsResult.Range("BO" & CStr(i)).value = FindStationInArray(aStation,
    wsResult.Range("BP" & CStr(i)).value)Next i
    ActiveWorkbook.Worksheets("Result").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Result").Sort.SortFields.Add Key:=Range( _
    "BO" & CStr(startRow) & ":BO" & CStr(endRow)), SortOn:=xlSortOnValues,
    Order:=xlAscending, DataOption:=xlSortNormal  '"Worksheet Result" กำหนดให้
    ลำดับการส่ง Column BO เป็น Index
    With ActiveWorkbook.Worksheets("Result").Sort  ' เรียงข้อมูลสินค้าทั้งหมด ตั้งแต่ แถวที่
    11.SetRange Range("BO" & CStr(startRow) & ":BV" & CStr(endRow))
    .Header = xlGuess .MatchCase = False
    Orientation = xlTopToBottom.SortMethod = xlPinYin
    .Apply End With startRow = endRow + 1 Next CarModifySolution
    wsResult.Range("BW8").Formula = "=SUM(BW11:BW10000)":
    wsResult.Range("BW8").value = wsResult.Range("BW8").value
    wsResult.Range("BW8").NumberFormat = "#,##0.00" End Sub
```

ประวัติผู้วิจัย

ชื่อ	นางณัตพร ไชยเสนา
วัน เดือน ปีเกิด	วันเสาร์ที่ 16 ธันวาคม 2532 จังหวัดกรุงเทพมหานคร
ประวัติการศึกษา	มหาวิทยาลัยศรีปทุม วิทยาศาสตร์มหาบัณฑิต สาขาวิชาการจัดการโลจิสติกส์และโซ่อุปทาน 2559
ตำแหน่งและสถานที่ทำงาน	อาจารย์ประจำสาขาวิชาการจัดการโลจิสติกส์ คณะบริหารธุรกิจ มหาวิทยาลัยราชพฤกษ์
ประสบการณ์ทำงาน	2554-2559 เลขานุการคณบดี คณะบริหารธุรกิจมหาวิทยาลัยราชพฤกษ์ 2559-ปัจจุบัน อาจารย์ประจำสาขาวิชาการจัดการโลจิสติกส์ คณะบริหารธุรกิจ มหาวิทยาลัยราชพฤกษ์

มหาวิทยาลัยราชพฤกษ์ RAJAPRUK UNIVERSITY