

ปัญหาการจัดการเส้นทางยานพาหนะทั่วไปที่มีขนาดของยานพาหนะและ
ขนาดของสินค้าที่หลากหลาย

ยุวดี ประกายเพ็ชรกุล

TNI

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

บัณฑิตวิทยาลัย สาขาเทคโนโลยีวิศวกรรม

สถาบันเทคโนโลยีไทย-ญี่ปุ่น

ปีการศึกษา 2557

A GENERALIZED VEHICLE ROUTING PROBLEM WITH MULTIPLE TRUCK SIZES AND
MULTIPLE PRODUCTS

Yuwadee Prakaiphetkul



A Thesis Submitted in Partial Fulfillment of Requirements
for The Degree of Master of Engineering Program in Engineering Technology

Graduate School

Thai-Nichi Institute of Technology

Academic Year 2014

หัวข้อวิทยานิพนธ์

ปัญหาการจัดการเส้นทางยานพาหนะทั่วไปที่มีขนาดของ
ยานพาหนะและขนาดของสินค้าที่หลากหลาย

โดย

ยุวดี ประกายเพชรกุล

สาขาวิชา

เทคโนโลยีวิศวกรรม

อาจารย์ที่ปรึกษาวิทยานิพนธ์

ผู้ช่วยศาสตราจารย์ ดร. พิศุทธิ์ พงศ์ชัยฤกษ์

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้บัณฑิตวิทยาลัยนี้เป็น
ส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร. พิชิต สุขเจริญพงษ์)

วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(ดร. บุญทริกา เกษมสันติธรรม)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. วิภาวดี วงษ์สุวรรณ)

.....กรรมการ

(ดร. กันติชา กิตติพิรชล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ผู้ช่วยศาสตราจารย์ ดร. พิศุทธิ์ พงศ์ชัยฤกษ์)

ยวดี ปรกายเพ็ชรกุล : ปัญหาการจัดการเส้นทางยานพาหนะทั่วไปที่มีขนาดของยานพาหนะ และขนาดของสินค้าที่หลากหลาย. อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร. พิศุทธิ์ พงศ์ชัยฤกษ์, 111 หน้า.

งานวิจัยนี้ศึกษาการขนส่งสินค้าของธุรกิจค้าปลีกรายหนึ่ง จากการศึกษาพบว่าการขนส่งสินค้าจากศูนย์กระจายสินค้าหนึ่งแห่งไปยังร้านค้าปลีกแต่ละแห่ง โดยใช้รถขนส่งมากกว่าหนึ่งประเภทซึ่งมีมาตรฐานความจุสินค้าที่แตกต่างกัน อีกทั้งขนาดของสินค้าที่ขนส่งนั้นยังมีความหลากหลายอีกด้วย ผู้วิจัยจึงนำปัจจัยต่าง ๆ มาประกอบการศึกษาการจัดการเส้นทางเดินรถ ให้ส่งสินค้าจากศูนย์กระจายสินค้าไปยังร้านค้าให้ได้ทั้งหมดด้วยต้นทุนการจัดส่งที่ต่ำที่สุด การศึกษาวิจัยครั้งนี้ได้ศึกษาวิธีการหาคำตอบด้วยวิธีการฮิวริสติกส์ โดยทำ Multiple Local Search ด้วย Neighborhood Search แบบวิธีเดียว และ Neighborhood Search แบบสองวิธีผสมกัน ซึ่งเกิดจาก SWAP INSERT และ REVERSE มาประกอบการจัดการเส้นทาง และแบ่งการพิจารณาเป็น 3 รูปแบบคือ กรณียอดส่งสินค้าสูง ยอดส่งสินค้าปานกลาง และยอดส่งสินค้าต่ำ พร้อมกับพิจารณาการขนส่งสินค้าแบบ Full Truckload (FTL) และ Less than Truckload (LTL) เปรียบเทียบกับค่าขนส่งที่เกิดขึ้นด้วยวิธีปัจจุบันที่ธุรกิจตัวอย่างดำเนินการอยู่ พบว่า วิธีการขนส่งที่เหมาะสม ได้ค่าขนส่งต่ำสุดตามวัตถุประสงค์ สำหรับการส่งสินค้าแบบ FTL คือ การปรับปรุงเส้นทางแบบ คือ SWAP และ REVERSE-SWAP สำหรับยอดส่งสินค้าสูงและปานกลาง สำหรับการส่งสินค้าแบบ LTL คือ การปรับปรุงเส้นทางแบบ คือ SWAP และ SWAP-INSERT สำหรับยอดส่งสินค้าสูงและปานกลาง แต่รูปแบบที่เหมาะสมกับยอดส่งสินค้าต่ำ คือ การไม่ปรับปรุงเส้นทาง ที่ระดับความเชื่อมั่นร้อยละ 90

บัณฑิตวิทยาลัย

สาขาวิชา เทคโนโลยีวิศวกรรม

ปีการศึกษา 2557

ลายมือชื่อนักศึกษา

ลายมือชื่ออาจารย์ที่ปรึกษา

YUWADEE PRAKAIPHETKUL : A GENERALIZED VEHICLE ROUTING PROBLEM WITH
MULTIPLE TRUCK SIZES AND MULTIPLE PRODUCTS. ADVISOR : ASSISTANCE
PROFESSOR DR. PISUT PONGCHAIRERKS, 111 PP.

In this study, goods distribution of a given retail business revealed that the delivery from a given distribution center to each store was operated by a number of types of trucks which had different standards of capacity. Furthermore, the sizes of goods are also varied. The researcher, therefore, studied delivery route planning which incurred lowest costs of transportation by taking into account various factors. Adopting Heuristic to solve the problem with multiple local search by single neighborhood search and double neighborhood search, this study compared three routing improvement techniques-SWAP, INSERT, and REVERSE-as well as two delivery methods-Full Truckload (FTL), and Less than Truckload (LTL) with business-as-usual (unimproved) delivery. However, this paper classifies demand in to three levels,: High Medium and Low. The results show that the delivery method for FTL is SWAP and REVERSE-SWAP for high and medium volume. The delivery method for LTL is SWAP and SWAP-INSERT for high and medium volume but the best for low volume is business-as-usual (unimproved) delivery with 90% significant level.

TNI

Graduate School

Field of Engineering of Technology

Academic Year 2014

Student's Signature.....

Advisor's Signature.....

กิตติกรรมประกาศ

วิทยานิพนธ์นี้กระทำให้สำเร็จลุล่วงไปด้วยดี อันเนื่องมาจากความกรุณาของ ผู้ช่วยศาสตราจารย์ ดร. พิศุทธิ์ พงศ์ชัยฤกษ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ กรุณาเสียสละเวลาอันมีค่าของท่าน มาให้คำแนะนำและให้คำปรึกษาแนวทางและการพัฒนาในการทำวิจัย ตลอดจนสอนขั้นตอนการดำเนินวิจัยตลอดการทำงานวิจัยนี้

ขอขอบพระคุณ ดร. บุญทริกา เกษมสันติธรรม ที่กรุณาเสียสละเวลาให้คำปรึกษา แนะนำและความรู้ในเรื่องต่างๆ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. วิภาวดี วงษ์สุวรรณ (สถาบันเทคโนโลยีไทย-ญี่ปุ่น) ที่กรุณาเสียสละเวลาเป็นคณะกรรมการสอบวิทยานิพนธ์ในครั้งนี้

ขอขอบพระคุณ ดร. กันติชา กิตติพิรชล (สถาบันเทคโนโลยีไทย-ญี่ปุ่น) ที่กรุณาเสียสละเวลาเป็นคณะกรรมการสอบวิทยานิพนธ์ในครั้งนี้

ขอขอบพระคุณ คุณอดิศักดิ์ อนุสรณ์สีรุ่ง ที่กรุณาเสียสละเวลาให้คำปรึกษาและคำแนะนำ ด้านการเขียนโปรแกรม

และท้ายที่สุดขอขอบพระคุณ คุณพ่อ คุณแม่ และครอบครัว ที่ให้โอกาสในการศึกษาและชอบคุณเพื่อนๆ อาจารย์ และผู้มีพระคุณทุกท่าน ทั้งผู้ที่กล่าวนามและไม่ได้กล่าวนามใน ณ ที่นี้ ผู้เขียนมีความซาบซึ้งในความกรุณาอันดีเยี่ยมจากทุกท่านและขอกราบขอบพระคุณมา ณ โอกาสนี้

ยุวดี ประกายเพชรกุล

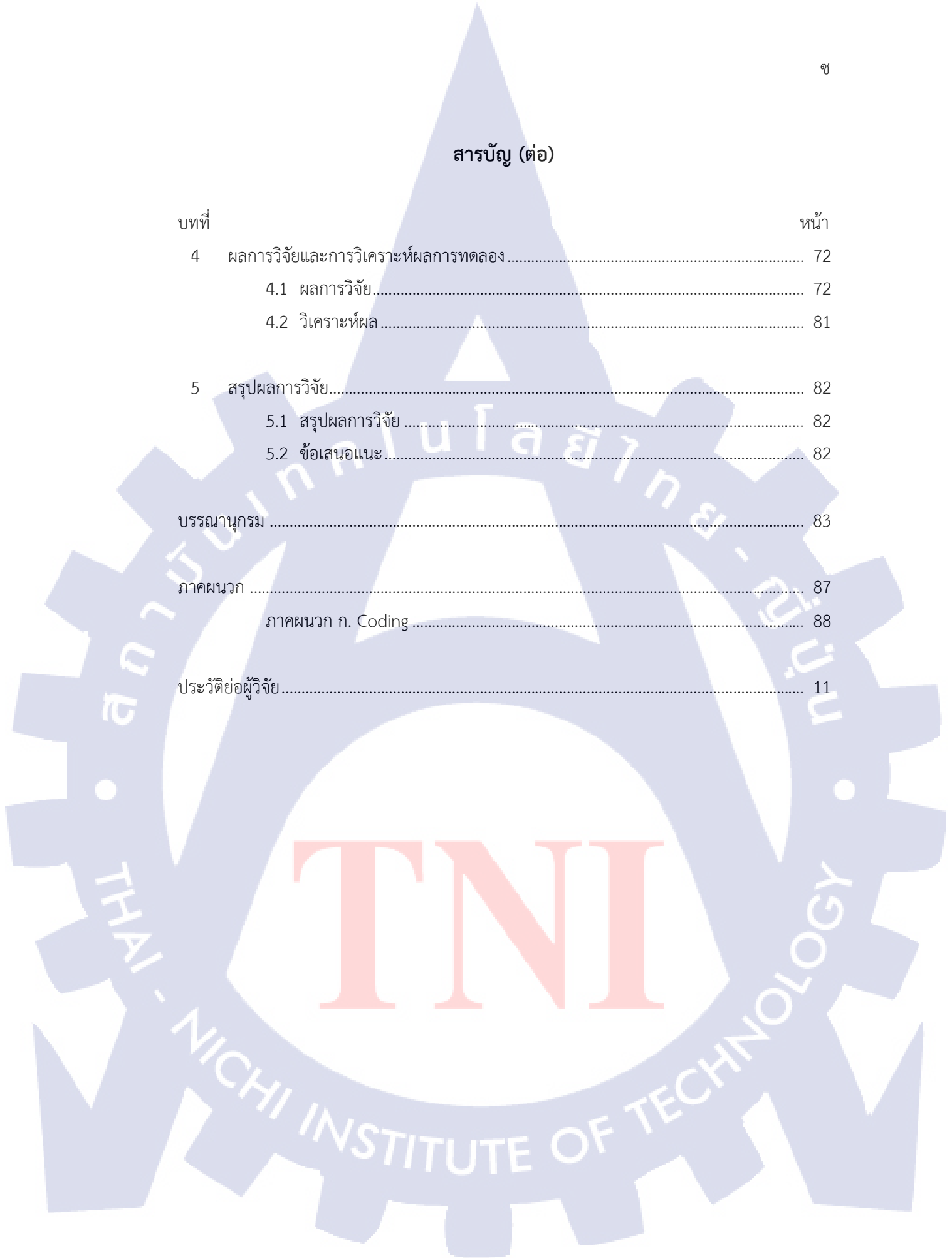
TNI

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญตาราง	ฌ
สารบัญรูป	ฎ
บทที่	
1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการศึกษา	3
1.3 ขอบเขตการวิจัย	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
1.5 สมมติฐานในงานวิจัย	3
1.6 แผนการดำเนินงาน	4
2 หลักการพื้นฐาน เอกสารและงานวิจัยที่เกี่ยวข้อง	5
2.1 ทฤษฎีที่เกี่ยวข้อง	5
2.2 ทบทวนวรรณกรรมการแก้ปัญหา VRP	17
3 ระเบียบวิธีวิจัย	23
3.1 ศึกษากระบวนการปัจจุบัน	23
3.2 กำหนดขอบเขตการศึกษา และเก็บข้อมูล	24
3.3 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง	24
3.4 สร้างโปรแกรมแบบจำลอง และทดสอบโปรแกรม	26

สารบัญ (ต่อ)

บทที่	หน้า
4 ผลการวิจัยและการวิเคราะห์ผลการทดลอง.....	72
4.1 ผลการวิจัย.....	72
4.2 วิเคราะห์ผล.....	81
5 สรุปผลการวิจัย.....	82
5.1 สรุปผลการวิจัย.....	82
5.2 ข้อเสนอแนะ.....	82
บรรณานุกรม.....	83
ภาคผนวก.....	87
ภาคผนวก ก. Coding.....	88
ประวัติย่อผู้วิจัย.....	11



TNI

สารบัญตาราง

ตาราง	หน้า
1.1 แผนการดำเนินงาน	4
2.1 ลักษณะปัญหาของ VRP	6
3.1 รูปแบบการคำนวณ	27
3.2 ข้อมูลและรูปแบบข้อมูลที่ใช้.....	27
3.3 ค่าเฉลี่ยยอดขายสินค้า ต่อร้าน ต่อวัน.....	28
3.4 จำนวนรถและขนาดบรรจุ.....	28
3.5 รูปแบบข้อมูลระยะทาง.....	28
3.6 รูปแบบข้อมูลค่าขนส่ง.....	29
4.1 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดขายสินค้าสูง และมีการบรรทุกแบบ FTL.....	72
4.2 เปรียบเทียบเวลาการคำนวณ (นาทีก) ของระดับยอดขายสินค้าสูง บรรทุกแบบ FTL.....	73
4.3 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดขายสินค้าปานกลาง และมีการบรรทุก แบบ FTL	73
4.4 เปรียบเทียบเวลาการคำนวณ (นาทีก) ของระดับยอดขายสินค้าปานกลาง และมีการบรรทุก แบบ FTL	74
4.5 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดขายสินค้าต่ำ และมีการบรรทุกแบบ FTL.....	75
4.6 เปรียบเทียบเวลาการคำนวณ (นาทีก) ของระดับยอดขายสินค้าต่ำ และมีการบรรทุก แบบ FTL	75
4.7 สรุปผลวิจัยที่ได้ สำหรับกรณีการบรรทุกสินค้าแบบ Full Truck Load.....	76
4.8 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดขายสินค้าสูง และมีการบรรทุกแบบ LTL.....	77
4.9 เปรียบเทียบเวลาการคำนวณ (นาทีก) ของระดับยอดขายสินค้าสูง และมีการบรรทุก แบบ LTL	77
4.10 เปรียบเทียบเวลาการคำนวณ (นาทีก) ของระดับยอดขายสินค้าสูง และมีการบรรทุก แบบ LTL	78
4.11 เปรียบเทียบเวลาการคำนวณ (นาทีก) ของระดับยอดขายสินค้าปานกลาง และมีการบรรทุก แบบ LTL	79
4.12 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดขายสินค้าปริมาณต่ำ และมีการบรรทุก แบบ LTL	79

สารบัญตาราง (ต่อ)

ตาราง	หน้า
4.13 เปรียบเทียบเวลาการคำนวณ (นาทื) ของระดับยอดสั่งสินค้าต่ำ และมีการบรรทุก แบบ LTL	80
4.14 สรุปผลวิจัยที่ได้ สำหรับกรณีการบรรทุกสินค้าแบบ Less than Truck Load.....	81



สารบัญรูป

รูป	หน้า
1.1 ลักษณะการจัดเส้นทางสำหรับยานพาหนะ.....	2
2.1 รูปแบบของ VRP	6
2.2 แผนผังวิธีการที่ใช้แก้ปัญหาที่มีชุดคำตอบที่เป็นไปได้แน่นอน	8
2.3 รูปแบบ Saving Algorithm	10
2.4 ตัวอย่างลักษณะการ SWAP	11
2.5 แผนภาพแสดงหลักการ SWAP	12
2.6 ตัวอย่างลักษณะการ INSERT	13
2.7 แผนภาพแสดงหลักการ INSERT	14
2.8 ตัวอย่างลักษณะการ REVERSE	15
2.9 แผนภาพแสดงหลักการ REVERSE	16
2.10 เทคนิคการรวมเส้นทาง ก่อนการคำนวณ Saving.....	17
2.11 การรวมเส้นทาง.....	19
3.1 แผนภาพแสดงกระบวนการทำวิจัย	23
3.2 รูปแบบการคำนวณการใช้รถแบบ Less than Truckload.....	25
3.3 รูปแบบการคำนวณการใช้รถแบบ Full Truckload.....	26
3.4 หลักการการปรับปรุงเส้นทางแบบ SWAP และคำนวณการใช้รถแบบ LTL	30
3.5 หลักการการปรับปรุงเส้นทางแบบ SWAP และคำนวณการใช้รถแบบ FTL	32
3.6 หลักการการปรับปรุงเส้นทางแบบ INSERT และคำนวณการใช้รถแบบ LTL.....	34
3.7 หลักการการปรับปรุงเส้นทางแบบ INSERT และคำนวณการใช้รถแบบ FTL.....	36
3.8 หลักการการปรับปรุงเส้นทางแบบ REVERSE และคำนวณการใช้รถแบบ LTL.....	38
3.9 หลักการการปรับปรุงเส้นทางแบบ REVERSE และคำนวณการใช้รถแบบ FTL.....	40
3.10 หลักการการปรับปรุงเส้นทางแบบ SWAP-INSERT และคำนวณการใช้รถแบบ LTL.....	42
3.11 หลักการการปรับปรุงเส้นทางแบบ SWAP-INSERT และคำนวณการใช้รถแบบ FTL.....	44
3.12 หลักการการปรับปรุงเส้นทางแบบ INSERT-SWAP และคำนวณการใช้รถแบบ LTL.....	46
3.13 หลักการการปรับปรุงเส้นทางแบบ INSERT-SWAP และคำนวณการใช้รถแบบ FTL.....	48
3.14 หลักการการปรับปรุงเส้นทางแบบ SWAP-REVERSE และคำนวณการใช้รถแบบ LTL	50
3.15 หลักการการปรับปรุงเส้นทางแบบ SWAP-REVERSE และคำนวณการใช้รถแบบ FTL	52

สารบัญรูป (ต่อ)

รูป	หน้า
3.16 หลักการการปรับปรุงเส้นทางแบบ REVERSE-SWAP และคำนวณการใช้รถแบบ LTL	54
3.17 หลักการการปรับปรุงเส้นทางแบบ REVERSE-SWAP และคำนวณการใช้รถแบบ FTL	56
3.18 หลักการการปรับปรุงเส้นทางแบบ INSERT-REVERSE และคำนวณการใช้รถแบบ LTL	58
3.19 หลักการการปรับปรุงเส้นทางแบบ INSERT-REVERSE และคำนวณการใช้รถแบบ FTL	60
3.20 หน้าจอหลัก	62
3.21 อธิบายส่วนประกอบของหน้าจอหลัก	63
3.22 อธิบายการใช้งานโปรแกรม รูปที่ 1	65
3.23 อธิบายการใช้งานโปรแกรม รูปที่ 2	66
3.24 อธิบายการใช้งานโปรแกรม รูปที่ 3	67
3.25 อธิบายการใช้งานโปรแกรม รูปที่ 4	68
3.26 อธิบายการใช้งานโปรแกรม รูปที่ 5	69
3.27 รูปแบบของการแสดงผลของ Export File รูปที่ 1	70
3.28 รูปแบบของการแสดงผลของ Export File รูปที่ 2	71
4.1 ปัจจัยที่มีอิทธิพลต่อปริมาณผลผลิต (Model 2) เปรียบเทียบกับปริมาณผลผลิต ที่เกษตรกรได้รับจริง และวิเคราะห์ร่วมกับชั้นความเหมาะสมของที่ดิน (Model 1)....	50

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันธุรกิจค้าปลีกและร้านสะดวกซื้อ เป็นธุรกิจที่เติบโตและมีการแข่งขันอย่างมากในประเทศไทย ซึ่งร้านสะดวกซื้อนี้เป็นช่องทางหนึ่งที่จะทำให้ลูกค้าสามารถได้รับสินค้าที่ต้องการอย่างสะดวกและรวดเร็ว ดังนั้น กระบวนการทางลอจิสติกส์จึงนับว่าเป็นส่วนสำคัญอย่างยิ่งในการดำเนินธุรกิจ (กวี ศรีเมือง. 2550) เพื่อให้สินค้าถึงมือลูกค้าได้ทันต่อความต้องการ ด้วยความหมายของลอจิสติกส์นั้น เป็นกระบวนการเกี่ยวกับการวางแผน การดำเนินการ และการควบคุม เพื่อให้ทรัพยากรเกิดการไหลอย่างมีประสิทธิภาพและครอบคลุมถึงประสิทธิภาพด้านต้นทุน การจัดเก็บสต็อก และเชื่อมโยงสารสนเทศ (โกศล ดีศีลธรรม. 2551) ปัจจัยสำคัญอย่างหนึ่งคือการขนส่ง ซึ่งไม่เพียงแต่ต้องมีความรวดเร็วในการส่งสินค้าไปร้านสาขาเท่านั้น ยังต้องมีความถูกต้องและครบถ้วน ภายใต้หลักเกณฑ์ด้านการลงทุนคือ อยู่ภายใต้ต้นทุนที่เหมาะสม

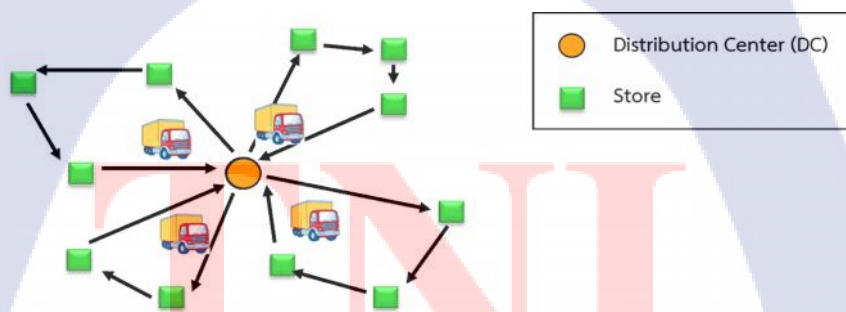
ธุรกิจร้านสะดวกซื้อหรือค้าปลีกเป็นธุรกิจหนึ่ง ที่ต้องอาศัยกระบวนการการขนส่งที่มีประสิทธิภาพ จากศูนย์กระจายสินค้าไปยังร้านสาขาทั้งหมดซึ่งปัจจุบันธุรกิจค้าปลีกที่ใหญ่ที่สุดในประเทศมีสาขามากที่สุดในประเทศไทย ประมาณ 7,200 สาขา (ข้อมูลเดือนสิงหาคม 2556)

ในการวางแผนเส้นทางหลักในการขนส่งสินค้าจากศูนย์กระจายสินค้าไปร้านสาขานั้น อาศัยโปรแกรมสำเร็จรูปในการหาเส้นทางสายหลัก ซึ่งเป็นเสมือนเส้นทางหลักที่ศูนย์กระจายสินค้านำไปใช้ แต่ในกรณีที่ร้านสาขาเปิดใหม่ในวันนั้นๆ ผู้ปฏิบัติหน้างานจะดำเนินการจัดร้านเพิ่มเข้าไปในแต่ละเส้นทางรถ ด้วยทักษะ ความชำนาญและประสบการณ์ส่วนตัว เนื่องจากร้านที่เพิ่มขึ้นมานั้น ไม่มีในเส้นทางรถสายหลักที่ประมวลผลมาจากโปรแกรมสำเร็จรูป ซึ่งนั่นเป็นข้อจำกัดบางประการของโปรแกรมสำเร็จรูปนี้ ทำให้ไม่สามารถนำมาใช้วางแผนเส้นทางพาหนะขนส่งประจำวันได้ โดยข้อจำกัดของโปรแกรมสำเร็จรูป คือ ใช้เวลาในการประมวลผลนาน (ประมาณ 4 ชั่วโมง สำหรับการดำเนินการปกติ ซึ่งมีการคำนวณจากปัจจัยตำแหน่งที่ตั้งร้าน, ระยะทางระหว่างศูนย์กระจายสินค้าและร้าน, ปริมาณการส่งสินค้าเฉลี่ยต่อวันรายร้าน) ซึ่งล่าช้าต่อการใช้งานในแต่ละวัน

ตารางที่ 1.1 ลักษณะของการขนส่งประจำวันของศูนย์กระจายสินค้าของธุรกิจ

ลักษณะการดำเนินการ	เงื่อนไข
1. จำนวนของยานพาหนะ	จำนวนหลายคัน (145 คัน)
2. ประเภทของยานพาหนะ	- 4 ล้อ (65 คัน) - 6 ล้อ (80 คัน)
3. ค่าใช้จ่ายรถขนส่ง	ต่างกันตามประเภทรถ (กำลังศึกษา)
4. ศูนย์กระจายสินค้า	จำนวน 1 ที่
5. ความต้องการในการขนส่ง	ความต้องการของร้านไม่แน่นอน
6. ความจุของรถบรรทุก	- 4 ล้อ ความจุ 5.5 คิว - 6 ล้อ ความจุ 10 คิว
7. ลักษณะสินค้า	2 ประเภท โดยคิดเป็นหน่วยเดียว (หน่วย : คิว) - Full case (ไปทั้งหีบสินค้า) - Break case (บรรจุลงลังพลาสติกมีฝาปิด)

ปัญหาลักษณะข้างต้นเป็นปัญหาการจัดเส้นทางสำหรับยานพาหนะ (Vehicle Routing Problem : VRP) กล่าวคือ เป็นการออกแบบเส้นทางเดินรถจากต้นทางหนึ่งจุดไปยังจุดปลายทางให้เหมาะสม ทั้งในด้านค่าใช้จ่ายหรือระยะทาง โดยมีข้อจำกัดต่างๆ เช่น ความจุรถบรรทุก เป็นต้น



รูปที่ 1.1 ลักษณะการจัดเส้นทางสำหรับยานพาหนะ

เช่นเดียวกับปัญหาที่ศึกษา ที่เป็นการออกแบบเส้นทางเดินรถที่เหมาะสมโดยมีข้อจำกัดด้านความจุซึ่งมีหลายขนาดความจุ(ตามประเภทรถ) และลักษณะของสินค้ามีหลายแบบ ทำให้ปัญหาที่ศึกษาเข้าข่ายปัญหา CVRP (Capacity Vehicle Routing Problem)

สำหรับปัญหานี้ นอกจากเป็นการศึกษากรณีที่มีรถหลายขนาด รถมีจำนวนจำกัด และมีมาตรฐานความจุของรถแล้ว ยังเป็นการศึกษากรณีที่เส้นทางเดินรถที่ออกจากศูนย์กระจายสินค้าและกลับมายังศูนย์กระจายสินค้ามีระยะทางไม่เท่ากัน (Asymmetric VRP) ลำดับในการจัดส่งสินค้าควรเป็นอย่างไร เพื่อให้ได้ค่าขนส่งที่ต่ำที่สุด

1.2 วัตถุประสงค์ของการศึกษา

เพื่อศึกษาการจัดลำดับการส่งสินค้าจากศูนย์กระจายสินค้าไปร้าน ให้มีค่าขนส่งน้อยที่สุด ภายใต้ ประเภทสินค้าที่หลากหลาย มีจำนวนรถที่จำกัด และมีมาตรฐานความจุของรถที่จำกัดตามประเภทรถ โดยอาศัยการคำนวณด้วยโปรแกรมอัตโนมัติ

1.3 ขอบเขตการวิจัย

1.3.1 จำนวนร้านค้าปลีก 896 สาขา

1.3.2 ประเภทรถขนส่งสินค้า มี 2 ประเภท

1.3.2.1 รถ 4 ล้อ ความจุ 5.5 คิว จำนวน 65 คัน

1.3.2.2 รถ 6 ล้อ ความจุ 10 คิว จำนวน 80 คัน

1.3.3 มีการส่งสินค้า 2 รูปแบบ (รวมใน 1 คัน) คือ

1.3.3.1 การส่งสินค้าแบบ Full Case คือ การส่งทั้งหีบสินค้า (หน่วย : คิว)

1.3.3.2 การส่งสินค้าแบบ Break Case คือ การส่งสินค้าแบบบรรจุลงลังมีฝาปิด

(หน่วย : คิว)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถนำไปประยุกต์ใช้กับการทำงาน จากวิธีการจัดลำดับการส่งสินค้าที่ลดภาระงานประจำวัน โดยคำนึงถึงต้นทุนค่าขนส่งต่ำที่สุด

1.4.2 ลดภาระงาน โดยอาศัยกระบวนการที่เป็นระบบในการทำงาน

1.5 สมมติฐานในงานวิจัย

1.5.1 รถขนส่ง 1 คัน สามารถบริการส่งได้หลายร้านสาขา แต่ 1 ร้านสาขาจะได้รับบริการส่งสินค้าโดยรถขนส่ง 1 คัน

1.5.2 ศึกษาการขนส่งสินค้าปกติ

1.5.3 เลือกใช้พาหนะขนาดใหญ่ก่อนพาหนะขนาดเล็ก

1.5.5 การคำนวณค่าขนส่งคิดเป็นโซน โดยวัดเป็นรัศมีระยะทางจากศูนย์กระจายสินค้า (ราคารวมน้ำมัน)

ประกอบด้วยขั้นตอนการศึกษาปัญหา วิเคราะห์ หาแนวทางและศึกษางานวิจัย จากนั้น
สร้างแบบจำลอง ทดลองโปรแกรมและสรุปผล และจัดทำบทความวิชาการและรูปเล่มงานวิจัย

[illegible]

บทที่ 2

หลักการพื้นฐาน เอกสารและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงหลักการพื้นฐาน เอกสารงานวิจัย และทฤษฎีที่เกี่ยวข้องกับการแก้ปัญหาด้านการขนส่งในงานวิจัยนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 ปัญหาการจัดการการขนส่ง

2.1.1.1 ปัญหาการจัดการขนส่ง

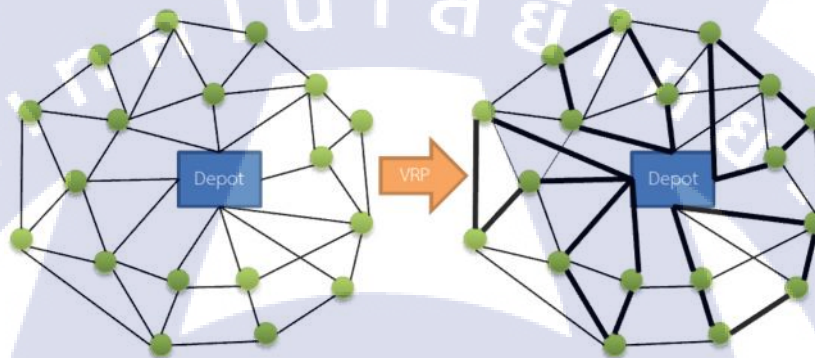
เป็นปัญหาที่เราต้องการคำตอบของการตัดสินใจที่ดีที่สุด ซึ่งอาจหมายถึงการมีประสิทธิภาพสูงสุด ผลกำไรสูงสุด หรือค่าใช้จ่ายในการดำเนินการน้อยที่สุด เป็นต้น การจำแนกปัญหาการตัดสินใจในธุรกิจสำหรับการขนส่ง คือ ปัญหาการตัดสินใจเชิงปฏิบัติการ เป็นการตัดสินใจที่ต้องการคำตอบเพื่อใช้ในการทำงาน ซึ่งอาจจะเป็นปัญหาการตัดสินใจที่ต้องการคำตอบที่ถูกต้องและแม่นยำอย่างมาก และปัญหาการตัดสินใจเชิงกลยุทธ์ เป็นปัญหาที่เกิดขึ้นตอนการวางแผนหรือการทดสอบนโยบายในการดำเนินการของบริษัท

2.1.1.2 ปัญหาการจัดเส้นทางสำหรับยานพาหนะ

สำหรับปัญหาการจัดเส้นทางยานพาหนะพัฒนามาจากปัญหาการเดินทางของพนักงานขาย (Traveling Salesman Problem : TSP) เป็นการหาเส้นทางที่สามารถบริการลูกค้าทั้งหมดที่กำหนดไว้ และกลับมาจุดเริ่มต้นโดยให้ระยะทางทั้งหมดน้อยที่สุด โดยปัญหาลักษณะนี้ ไม่มีข้อจำกัดด้านความจุเกี่ยวข้องกับ ส่วนของปัญหาการจัดเส้นทางยานพาหนะ (Vehicle Routing Problem : VRP) คือการกำหนดกลุ่มของเส้นทางขนส่งจากคลังสินค้าไปยังความต้องการจุดต่างๆ โดยที่ค่าใช้จ่ายหรือระยะทางในการขนส่งมีค่าน้อยที่สุด ภายใต้ข้อจำกัด เช่น เวลา หรือขีดจำกัดของยานพาหนะ เป็นต้น โกลเด้น บี; ที แมกนันตี; และเอช จุนเยน (Golden B; T. Magnanti; and H. Nguyen. 1977) ได้เสนอปัญหาการจัดเส้นทางยานพาหนะจากคลังสินค้าไปยังลูกค้าหลายจุด โดยมีเงื่อนไขคือ ระยะทางต่ำที่สุด และทุกๆ ยานพาหนะจะเริ่มต้นและสิ้นสุดที่คลังสินค้าโดยมีข้อจำกัดด้านการบรรทุกและเวลาสูงสุดในการขนส่งหนึ่งรอบการจัดส่ง ถ้าไม่คำนึงถึงข้อจำกัดด้านเวลาสูงสุด จะเป็นปัญหาการจัดเส้นทางยานพาหนะมาตรฐาน หรือ Standard Vehicle Routing Problem หรือ VRP (อุบลรัตน์ เจริญนาคม. 2551) นั้นหมายถึง รูปแบบปัญหานี้มีการพิจารณาด้านความจุเข้ามาเกี่ยวข้อง ปัญหาการจัดเส้นทางยานพาหนะ นำเสนอโดย Dantzing G.B. and Ramser J.H. (1959 : 80-91) จุดประสงค์ของ VRP คือ การจัดส่งถึงลูกค้าตามความต้องการ

ของลูกค้าโดยต้องมีค่าใช้จ่ายในเส้นทางการขนส่งต่ำที่สุด ซึ่งมีการแบ่งระดับการวางแผนการจัดการขนส่งได้เป็น 3 ระดับ ดังนี้

1. การวางแผนเชิงกลยุทธ์ เป็นส่วนของการวางแผนที่เกี่ยวข้องกับทิศทางในการบริหาร เช่น การกำหนดที่ก่อสร้าง ขอบเขตการให้บริการ เป็นต้น
2. การวางแผนเชิงควบคุม เป็นส่วนของการควบคุมซึ่งเป็นหนึ่งในการบริหาร เช่น ควบคุมจำนวนพาหนะ เป็นต้น
3. การวางแผนเชิงปฏิบัติการ เป็นส่วนของการจัดการเส้นทางสำหรับพาหนะ



รูปที่ 2.1 รูปแบบของ VRP

ตารางที่ 2.1 ลักษณะปัญหาของ VRP

ลักษณะของปัญหา	ทางเลือก
1. จำนวนของยานพาหนะ	จำนวน 1 คัน / จำนวนหลายคัน
2. ประเภทของยานพาหนะ	ประเภทเดียว / หลายประเภท
3. ศูนย์กระจายสินค้า	จำนวน 1 ที่ / จำนวนหลายที่
4. ความต้องการในการขนส่ง	ความต้องการแน่นอน / ความต้องการไม่แน่นอน
5. จุดกำเนิดของความต้องการ	ที่ตำแหน่ง / ที่เส้นทาง / ที่ตำแหน่งและเส้นทาง
6. ความจุรถบรรทุก	เท่ากัน / ไม่เท่ากัน
7. เวลาในการขนส่งที่ยอมให้มากที่สุด	เท่ากัน / ไม่เท่ากัน
8. ข้อจำกัดด้านเวลาในการขนส่ง	เท่ากัน / ไม่เท่ากัน

ที่มา : กวี ศรีเมือง. (2550). การหาจำนวนพาหนะบรรทุกที่เหมาะสมในการขนส่งสินค้าในธุรกิจค้าปลีก กรณีศึกษาที่อปัสซูเปอร์มาร์เก็ต. หน้า 17.

นอกจากนั้นแล้ว VRP ยังแบ่งประเภทตามลักษณะปัญหาที่มีเงื่อนไข ได้ดังนี้

- Vehicle Routing Problem with Pickup and Delivery (VRPPD) จำนวนของสินค้าถูกส่งจากสถานที่หนึ่งไปอีกสถานที่หนึ่ง โดยมีเส้นทางที่เหมาะสมที่สุดสำหรับเส้นทางรถในการไปรับและส่งสินค้า

- Vehicle Routing Problem with Time Windows (VRPTW) มีข้อกำหนดเรื่องช่วงเวลาในการขนส่งสินค้า

- Capacity Vehicle Routing Problem (with or without Time Windows) (CVRP or CVRPTW) มีข้อจำกัดการบรรทุกสินค้าที่ต้องส่ง

- Vehicle Routing Problem with Multiple Trips (VRPMT) ยานพาหนะสามารถเดินทางได้มากกว่า 1 เส้นทาง

2.1.1.3 วิธีหาคำตอบสำหรับปัญหาการจัดเส้นทางยานพาหนะ

a) วิธีการหาคำตอบที่ดีที่สุด

เป็นวิธีการหาคำตอบที่ดีที่สุด และไม่ขัดแย้งกับทุกเงื่อนไขที่กำหนด แต่ใช้เวลานานเมื่อมีเงื่อนไขมาก หรือมีตัวแปรที่เพิ่มขึ้น ซึ่งมีวิธีที่นิยมใช้ คือ Branch and Bound (Up to 100 Node) และ Branch and Cut เป็นต้น

b) วิธีการหาคำตอบที่ดีที่สุด

วิธีฮิวริสติกส์ เป็นวิธีการหาคำตอบที่ดีที่สุด ในขณะที่ขณะนั้น ภายในระยะเวลาและไม่ขัดแย้งกับเงื่อนไขที่กำหนด เพราะเนื่องจากในบางครั้งปัญหามีตัวแปรและเงื่อนไขที่ซับซ้อนมาก ไม่อาจเขียนให้อยู่ในรูปแบบจำลองทางคณิตศาสตร์ได้ ซึ่งมีวิธีที่นิยมใช้ เช่น Savings Algorithm (Clarke; and Wright. 1963). Matching Based เป็นต้น นอกจากนี้แล้ว ยังมีการสร้างฮิวริสติกประเภทหนึ่งที่เรียกว่า เมตาฮิวริสติก ซึ่งหมายถึงชุดของลำดับขั้นตอนการแก้ปัญหา (Algorithm) แบบฮิวริสติกชนิดหนึ่งที่สามารถนำหลักการเดียวกันไปใช้แก้ปัญหาได้หลากหลายปัญหา ซึ่งในปัจจุบัน วิธีการออกแบบฮิวริสติกโดยอาศัยหลักการทางเมตาฮิวริสติกนี้ได้รับความนิยมเช่นกัน เนื่องจากคำตอบที่ได้ให้ผลที่ดี แก้ปัญหาได้เร็วและใช้งานได้ง่ายเป็นดังนี้

- เมตาฮิวริสติกมีระเบียบวิธีในการค้นหาคำตอบที่ดีภายในพื้นที่ของคำตอบที่เป็นไปได้

- เมตาฮิวริสติกมีวัตถุประสงค์เพื่อหาคำตอบที่ดีที่สุดหรือคำตอบที่ใกล้เคียงคำตอบที่ดีที่สุดภายในระยะเวลาอันสั้น

- วิธีการทางเมตาฮิวริสติกอาจมีทั้งแบบง่ายไม่ซับซ้อน เช่น การปรับปรุงคำตอบเฉพาะที่ (Local Search) หรือแบบที่ยุ่งยากกว่า คือวิธีระบบมด (Ant System) วิธีการเชิงพันธุกรรม (Genetic Algorithm) วิธีการค้นหาต้องห้าม (Tabu Search) วิธีการเลียนแบบการอบอ่อน (Simulated Annealing) เป็นต้น

- เมตาฮิวริสติกเป็นขั้นตอนการประมาณคำตอบ

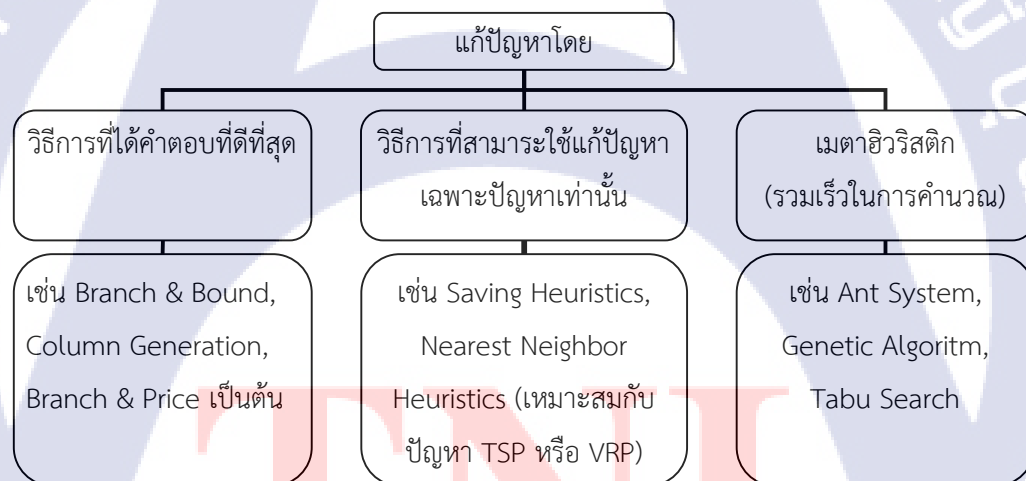
- เมตาฮิวริสติกอาจเกิดจากการรวมหลากหลายเทคนิค เพื่อค้นหาคำตอบที่ดีที่สุดภายในพื้นที่คำตอบที่เป็นไปได้

- เมตาฮิวริสติกมีระเบียบขั้นตอนมาตรฐานที่แน่นอน

- เมตาฮิวริสติกต้องสามารถใช้ได้กับหลากหลายปัญหา

- เมตาฮิวริสติกอาจเป็นการบรรยาย หรือไม่จำเป็นต้องมีหลักการทางคณิตศาสตร์

ซึ่งในปัจจุบัน มีวิธีเมตาฮิวริสติกอยู่หลากหลาย เช่น Ant System, Genetic Algorithm, Tabu Search, Trajectory Method, Local Search เป็นต้น



รูปที่ 2.2 แผนผังวิธีการที่ใช้แก้ปัญหาที่มีชุดคำตอบที่เป็นไปได้แน่นอน

ที่มา : ระพีพันธุ์ ปีตาอะโส. (2554). วิธีการเมตาฮิวริสติก **Metaheuristic** เพื่อแก้ไขปัญหการวางแผนการผลิตและการจัดการโลจิสติกส์. หน้า 16.

2.1.1.4 วิธีสำหรับการแก้ปัญหาการจัดเส้นทางยานพาหนะ

2.1.1.4.1 การจัดกลุ่มรับสินค้าก่อนแล้วจัดเส้นทางยานพาหนะ

เป็นการจัดเส้นทางโดยกำหนดกลุ่มลูกค้าก่อน เช่น Petal Algorithm, Sweep Algorithm เป็นต้น

2.1.1.4.2 การจัดเส้นทางยานพาหนะก่อนแล้วจัดกลุ่มรับสินค้า

เป็นการจัดเส้นทางเชื่อมกันที่เหมาะสมระหว่างลูกค้าก่อน แล้วจึงแบ่งเส้นทางที่ได้เป็นเส้นทางย่อย จากเงื่อนไขต่างๆ เช่น ความจุของรถ ความต้องการของลูกค้า เป็นต้น เทคนิคที่ใช้สำหรับกรณีนี้ เช่น Saving Method, Nearest Neighbor Method เป็นต้น

- Nearest Neighbor Method เป็นการจัดเส้นทางยานพาหนะจากการพิจารณาระยะทางระหว่างแต่ละจุด ให้เชื่อมต่อกันโดยมีระยะทางระหว่างจุดใกล้เคียงกันมากที่สุด

- Saving Method จาก คลาก; และไรท์ (Clarke; and Wright. 1963 : 568-581) ได้ศึกษาเกี่ยวกับเส้นทางที่เหมาะสมที่สุดสำหรับรถขนส่งที่มีความจุขนส่งต่างกันจากศูนย์กระจายสินค้าไปยังสถานที่ต่างๆ หลักการคือ กำหนดให้รถวิ่งระหว่างศูนย์กระจายสินค้าและลูกค้าจุดต่อจุดแบบไปกลับ และกำหนดให้รถวิ่งระหว่างศูนย์กระจายสินค้าและลูกค้ามากกว่าหนึ่งจุด ซึ่งสิ่งที่ได้จากการแก้ปัญหา นี้ จะได้ระยะทางที่สั้นที่เหมาะสมที่สุดจัดเส้นทางยานพาหนะ

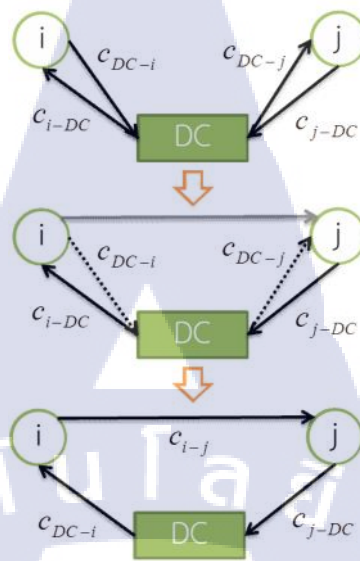
ซึ่งมีสมการดังนี้ $S_{i-j} = C_{i-DC} + C_{DC-j} - C_{i-j}$

โดยที่ S_{i-j} = ค่าความประหยัดของระยะเวลา หรือ ระยะทาง หรือ ค่าใช้จ่าย ระหว่างจุด i และจุด j

C_{i-DC} = ระยะเวลา หรือ ระยะทาง หรือ ค่าใช้จ่าย ระหว่างจุด i และ DC

C_{DC-j} = ระยะเวลา หรือ ระยะทาง หรือ ค่าใช้จ่าย ระหว่าง DC และจุด j

C_{i-j} = ระยะเวลา หรือ ระยะทาง หรือ ค่าใช้จ่าย ระหว่างจุด i และจุด j



รูปที่ 2.3 รูปแบบ Saving Algorithm

มีขั้นตอนการดำเนินการ ดังต่อไปนี้

1. สร้างคำตอบเริ่มต้น (Initial Solution)
2. คำนวณค่าความประหยัดของระยะเวลา หรือ ระยะทาง หรือ ค่าใช้จ่าย
3. เรียงลำดับค่าความประหยัด S_{ij} จากมากไปน้อย
4. รวมเส้นทางของจุด i และจุด j ให้อยู่ในเส้นทางเดียว
5. ทำซ้ำจนจัดเส้นทางยานพาหนะได้ครอบคลุมจุดทั้งหมด โดยมีเงื่อนไขการบรรทุก

ไม่เกินความจุของยานพาหนะ (กวี ศรีเมือง. 2550)

2.1.1.5 รูปแบบการขนส่งสินค้าด้วยยานพาหนะ

การจัดส่งสินค้าในปัจจุบันนี้มีหลายรูปแบบ สำหรับรูปแบบที่นำมาศึกษา

คือ Less Than Truckload และ Full Truckload

2.1.1.5.1 Less than Truck Load (LTL)

เป็นการจัดส่งสินค้าแบบไม่เต็มคันรถ ในกรณีนี้ต้นทุนจะสูง แต่การดำเนินการขนส่งง่ายกว่าเนื่องจากไม่ต้องรอหรือรวบรวมสินค้าให้เต็มคัน

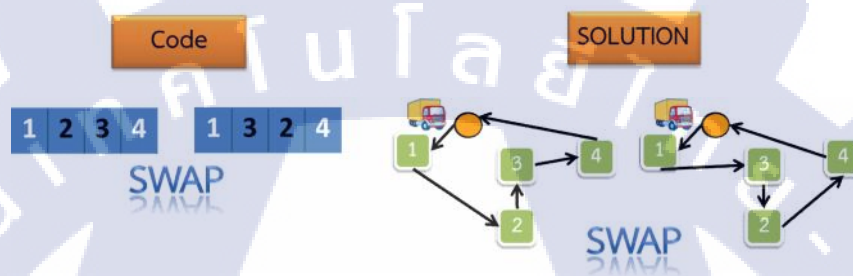
2.1.1.5.2 Full Truck Load (FTL)

การขนส่งตรงแบบเต็มคันรถ เป็นแนวคิดหรือกลยุทธ์ที่นำมาใช้ลดต้นทุนการขนส่งสินค้าที่มีมานาน แต่ไม่เป็นที่นิยมมากนักเนื่องจากทำได้ยาก ต้องอาศัยกระบวนการทำงานขององค์กรเป็นอย่างดีรวมถึงลูกค้า ข้อดีของการขนส่งตรงแบบเต็มคัน คือ

รวดเร็วและมีต้นทุนต่ำ แต่การที่จะต้นทุนต่ำได้นั้น สินค้าจะต้องเต็มคันรถ ถ้าสินค้าไม่เต็มคันรถ ต้นทุนขนส่งก็จะสูง รวมถึงต้องคำนึงความถี่ในการส่งมอบสินค้าด้วย (Logistics Corner, 2553)

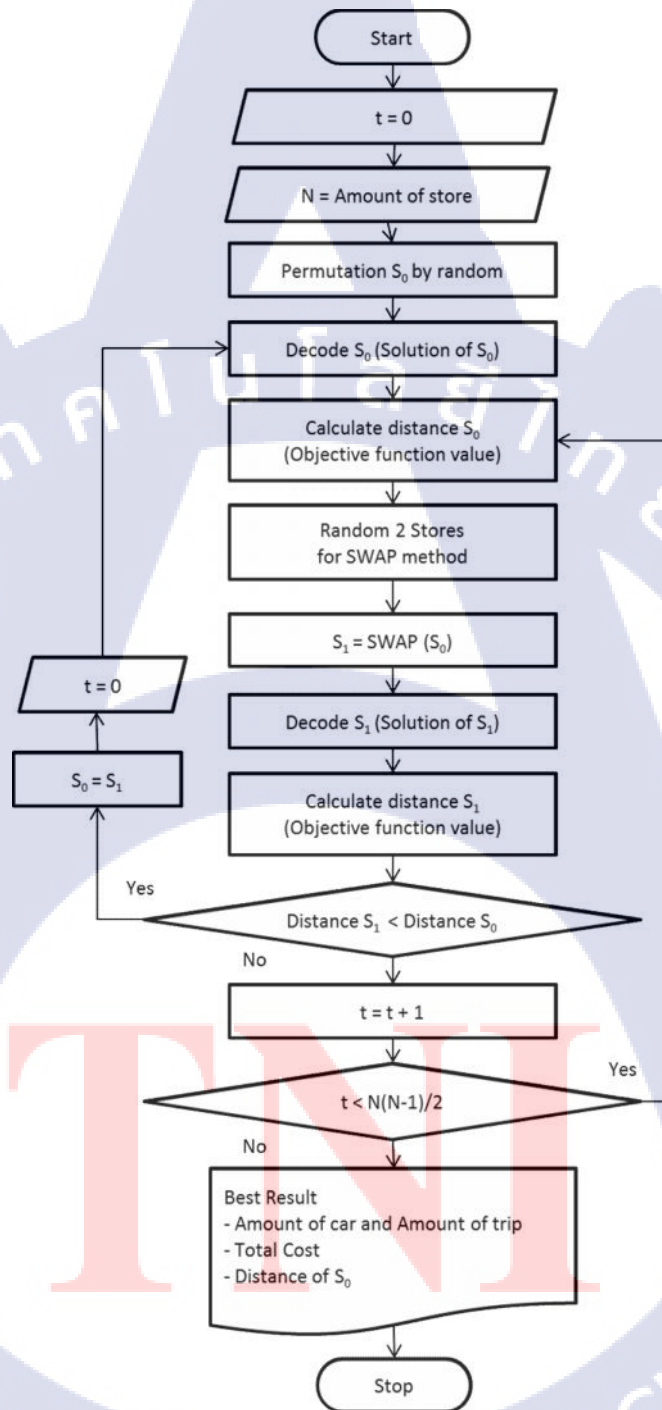
สำหรับงานวิจัยนี้ ได้นำวิธี Nearest Neighbor Method มาประกอบ การศึกษา และใช้เทคนิคการปรับปรุงเส้นทาง 3 วิธีคือ SWAP, INSERT และ REVERSE

รูปแบบการ SWAP เป็นวิธีการหนึ่งใน Neighborhood Search ซึ่งเป็นการสับเปลี่ยนตำแหน่ง 2 ตำแหน่ง i และ j โดยที่ $1 \leq i \leq j \leq n$ เมื่อ i, j คือตำแหน่งในการสับเปลี่ยน และ n คือ จำนวนตำแหน่งทั้งหมด



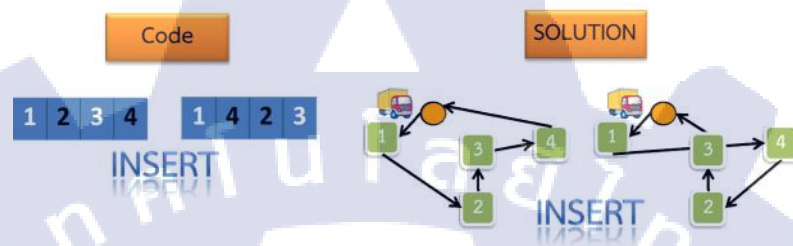
รูปที่ 2.4 ตัวอย่างลักษณะการ SWAP

โดยมีหลักการทำงานเป็นดังนี้



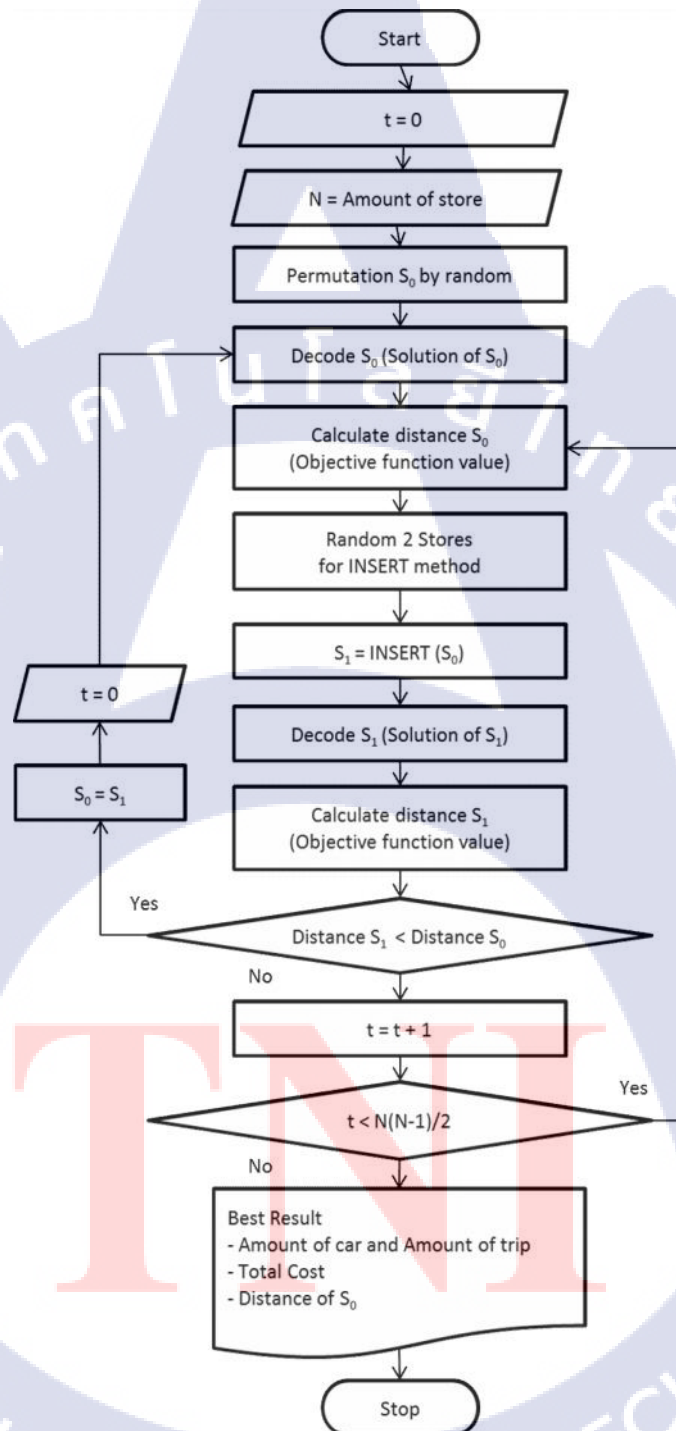
รูปที่ 2.5 แผนภาพแสดงหลักการ SWAP

รูปแบบการ INSERT ดำเนินการโดยเลือกจุดที่จะนำไปแทรก (ขั้นตอนเลือก) และตัดสินใจนำไปแทรกที่ตำแหน่งซึ่งไม่ใช่ตำแหน่งที่ใกล้กัน (ขั้นตอนแทรก) และสามารถแบ่งการ INSERT ได้เป็นหลายรูปแบบคือ Nearest insertion, Cheapest Insertion, Arbitrary Insertion, Farthest insertion, Quick Insertion, Convex Hull Insertion, Greatest Angle Insertion และ Difference x Ratio Insertion (L. Bodin; et al. 1983)



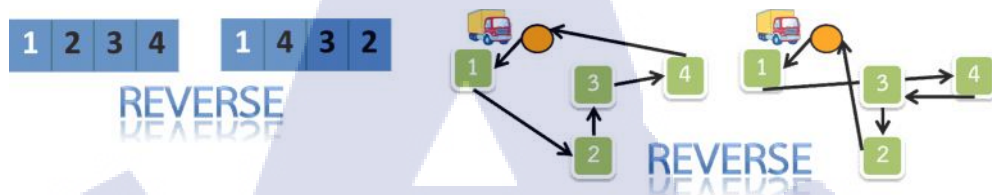
รูปที่ 2.6 ตัวอย่างลักษณะการ INSERT

โดยมีหลักการทำงานเป็นดังนี้



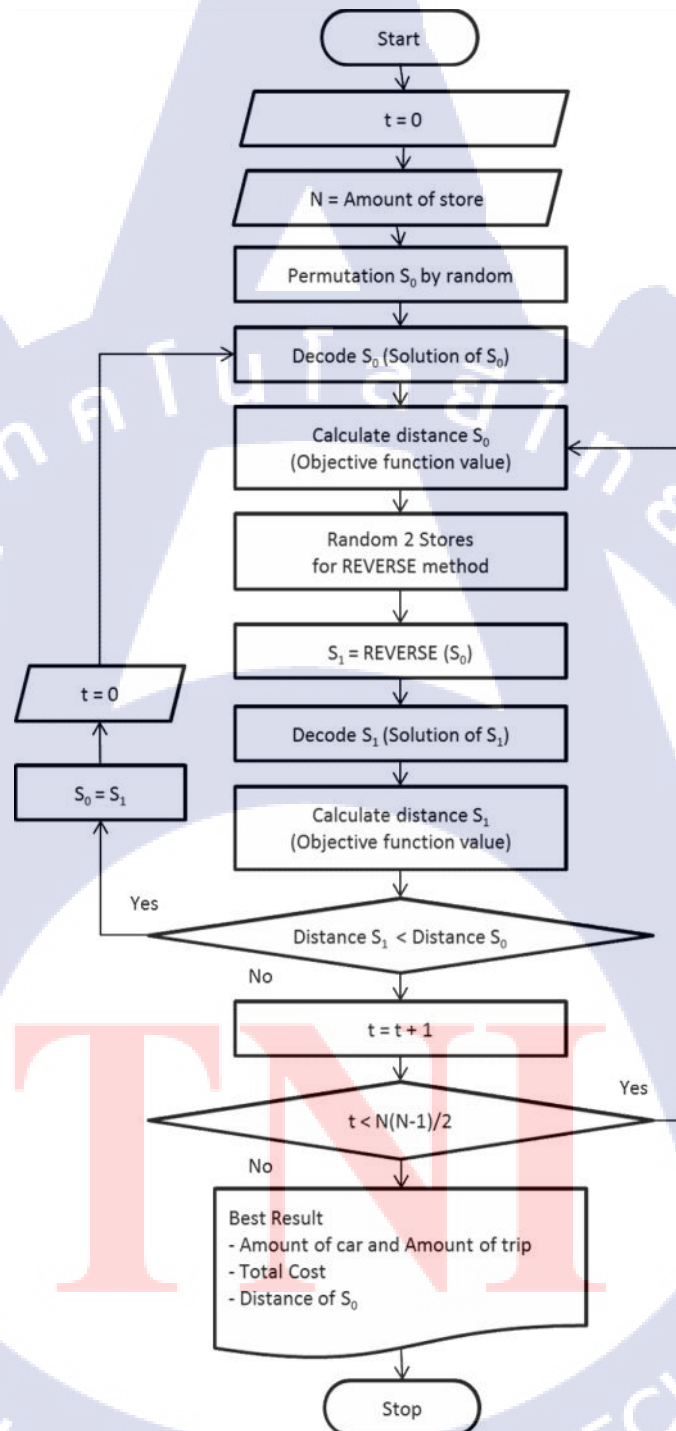
รูปที่ 2.7 แผนภาพแสดงหลักการ INSERT

รูปแบบการ REVERSE เป็นการพิจารณาจากลำดับเดิมและดำเนินการกลับลำดับ ตั้งแต่ลำดับที่ i ถึง j โดยที่ $1 \leq i \leq j \leq n$ เมื่อ i, j คือตำแหน่งหัวท้ายของการกลับลำดับ และ n คือจำนวนตำแหน่งทั้งหมด



รูปที่ 2.8 ตัวอย่างลักษณะการ REVERSE

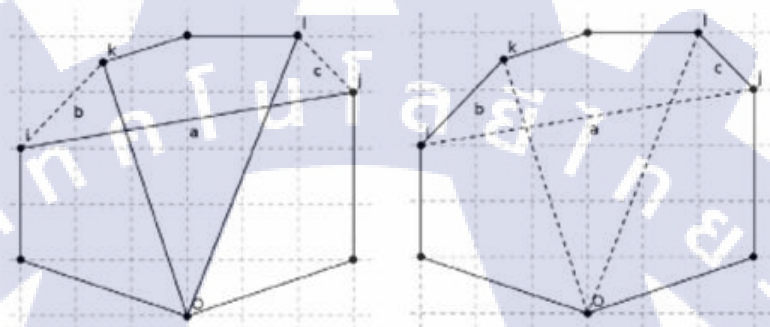
โดยมีหลักการทำงานเป็นดังนี้



รูปที่ 2.9 แผนภาพแสดงหลักการ REVERSE

2.2 ทบทวนวรรณกรรมการแก้ปัญหา VRP

มีแลน สตาร์โนจิวิก; บ็องดาน่า สตาร์โนจิวิก; และไมโก้ วูโจชีวิก (Milan Stanojevic; Bogdana Stanojevic; and Mirko Vujošević. 2013) ศึกษาการปรับปรุงการคำนวณ Saving สำหรับปัญหา CVRP ซึ่งส่วนใหญ่วิธีการที่นิยมในการแก้ปัญหา VRP จะใช้ Clarke-Wright Saving Algorithm แต่ในงานวิจัยนี้ ผู้วิจัยได้แนะนำการคำนวณ Saving ในแบบการ Merge เส้นทาง และใช้วิธี Extended Saving Algorithm (ESA)



(a) Before Merging

(b) After Merging

รูปที่ 2.10 เทคนิคการรวมเส้นทาง ก่อนการคำนวณ Saving

ที่มา : Milan Stanojevic; Bogdana Stanojevic; and Mirko Vujošević. (2013). Enhances Saving Calculation and its Applications for Solving Capacitated Vehicle Routing Problem. p. 10,302-10,312.

- (a) ก่อนรวมเส้นทาง: มี 2 เส้นทาง คือ $q = (0, \dots, i, j, \dots, 0)$ และ $r = (0, k, \dots, l, 0)$
- (b) หลังรวมเส้นทาง: มี 1 เส้นทาง คือ $s = (0, \dots, i, k, \dots, l, j, \dots, 0)$ โดยเส้น (i, j) ได้ถูกแทนที่เส้นทางด้วย $(i, k, \dots, l, j)$ และสำหรับ $(0, k)$ และ $(l, 0)$ ซึ่งก็ได้ถูกแทนที่เส้นทางด้วย (i, k) และ (l, j) เช่นกัน

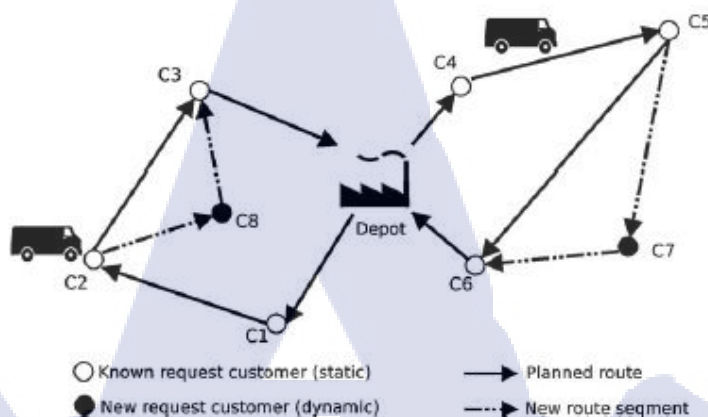
ในงานวิจัยได้กล่าวว่า ทั้ง Clarke-Wright Saving Algorithm และ Extended Saving Algorithm มีการคำนวณที่ซับซ้อนเหมือนกัน สำหรับ CW Algorithm คำนวณ Saving ในขั้นตอนแรก และลำดับมาเรื่อยๆ จนครบรอบซ้ำ แต่สำหรับ ESA Algorithm นั้น การคำนวณ Saving จะเกิด

หลังจากการ Merge และโดยสรุป ผลการวิจัยพบว่า ค่าเฉลี่ยของการคำนวณแบบ ESA มีค่าที่ดีกว่า ในแบบการคำนวณเดิม

แรน รียู; เซาหลัน ซี; วองเซนต์ ออกัสโต; และคาร์ลอส รอดดริเกซ (Ran Liu; Xiaolan Xie; Vincent Augusto; and Carlos Rodriguez. 2013) ศึกษาการส่งสินค้าแบบที่จุดรับและจุดส่ง เป็นจุดเดียวกัน อีกทั้งมีข้อจำกัดด้านเวลา ซึ่งเป็นรูปแบบปัญหาที่เกี่ยวข้องกับ Vehicle Routing Problem with Simultaneous Delivery and Pickup and Time Windows (VRPSDPTW) โดยผู้วิจัยได้ศึกษากับธุรกิจ Home Health care เนื่องจากผู้วิจัยเห็นว่าเป็นปัญหาที่ซับซ้อน และยังเป็นปัญหาสำคัญที่เกิดขึ้นกับธุรกิจ จุดประสงค์เพื่อให้มีต้นทุนยานพาหนะต่ำที่สุดที่เพียงพอต่อความต้องการของคนไข้ โดยใช้เมตะฮิวริสติกส์ 2 เรื่องคือ Tabu Search (TS) และ Genetic Algorithm (GA) ในการแก้ไขปัญหา ผู้วิจัยได้นำ GA มาประกอบการเปลี่ยนลำดับและปรับปรุงด้วย Local Search เนื่องด้วยจุดของคนไข้เป็นแบบทั้งรับ-ส่งสินค้า วิธีนี้จึงเป็นวิธีที่เหมาะสม (Prins, 2004) อีกส่วนหนึ่งในการแก้ปัญหา VRP ผู้วิจัยได้เลือก Tab Search เนื่องจากเป็นเทคนิคที่เหมาะสมกับปัญหานี้ (Cote and Potvin, 2009) โดยการย้ายคนไข้จากเส้นทางรถหนึ่งไปอีกเส้นทางรถอื่น เพื่อให้ได้เส้นทางที่ดีที่สุดตามจุดประสงค์ และเนื่องจากเป็นปัญหาใหม่จึงยังไม่มีเกณฑ์สำหรับเปรียบเทียบ แต่อาศัยคิดเปรียบเทียบกับ VRPTW โดยสรุปงานวิจัย พบว่าวิธีใหม่ที่คิดขึ้นมานี้ได้ผลดีกว่า อีกทั้งยังสามารถนำไปวางแผนการขนส่งได้

ชัยยศ ผิวปานแก้ว; และอนันต์ มุ่งวัฒนา (2556) ศึกษาเกี่ยวกับปัญหาการขนส่งในสถานการณ์จริง ซึ่งลูกค้าถูกจัดอยู่ในตำแหน่งเขตที่แตกต่างกัน โดยการจัดส่งนั้นอาศัยทักษะและความชำนาญของผู้จัดตารางขนส่ง ดังนั้นผู้วิจัยจึงได้พัฒนาแบบจำลองทางคณิตศาสตร์และอัลกอริทึมใหม่ และทำการเปรียบเทียบอัลกอริทึมแบบเก่าและใหม่ โดยสรุปการศึกษานี้ สามารถลดค่าใช้จ่ายการขนส่งได้ประมาณ 5% แต่มีข้อเสียเรื่องเวลาการใช้หาคำตอบที่นาน

มอสเทฟา อาร์; และคณะ (Mostepha R; and et al. 2011) เปรียบเทียบการศึกษาปัญหา Vehicle Routing Problem with Dynamic Requests ด้วย PSO (Particle Swarm Optimization) และ Variable Neighborhood Search (VNS) ผู้วิจัยได้กล่าวว่า VRP เป็นปัญหาสำคัญทั้งในงานค้นคว้าวิจัยและอุตสาหกรรมขนส่งที่มีอยู่ ซึ่งเป็นที่ทราบกันดีว่า VRP เป็นปัญหาในการออกแบบเส้นทางรถเพื่อให้บริการปลายทางโดยให้ต้นทุนต่ำที่สุด งานวิจัยนี้จึงนำปัญหานี้มาศึกษา ในกรณีที่มีความต้องการจากลูกค้าไม่แน่นอน หรือไม่ทราบก่อนการจัดเส้นทาง ตัวอย่างเช่น ความต้องการและตำแหน่งของลูกค้ามาหลังจากที่จัดเส้นทางรถเรียบร้อยแล้ว ดังตัวอย่างในภาพ จะเห็นว่าที่วงกลมสีดำทึบเป็นความต้องการของลูกค้าที่เข้ามาใหม่ และนั่นเป็นส่วนหนึ่งที่ต้องดำเนินการจัดเส้นทางรถใหม่



รูปที่ 2.11 การรวมเส้นทาง

ที่มา : Mostepha R; et al. (2011). A Comparative Study Between Dynamic Adapted PSO and VNS for the Vehicle Routing Problem with Dynamic Requests. p. 1,426-1,439.

โดยสรุปแล้ว เมื่อเปรียบเทียบระหว่าง PSO และ VNS มีผลต่างกันเล็กน้อย โดย PSO มีประสิทธิภาพการคำนวณที่ดี ขณะที่ VNS คำนวณระยะทางเฉลี่ยได้สั้นกว่า เมื่อวัดผลในเรื่องประสิทธิภาพของ Algorithm นั้นพบว่า VNS มีความถูกต้องมากกว่า PSO ในขณะที่ PSO ให้ผลลัพธ์ที่คงที่

ฐิตินันท์ ศรีสุวรรณณี; และรพีพันธ์ พิทักษ์โส (2555) ศึกษาเกี่ยวกับการแก้ปัญหาการจัดเส้นทางขนส่งด้วยวิธีการ Ant Colony Optimization ผู้วิจัยออกแบบอัลกอริทึมโดยประยุกต์จาก Ant Colony Optimization และปรับปรุงคำตอบด้วยวิธี Crossover-Move การสลับสองตำแหน่ง (2-Opt) และการย้ายหนึ่งตำแหน่ง (One-Move) เพื่อแก้ปัญหา VRP ของกรณีศึกษาซึ่งมีข้อจำกัดของความจุยานพาหนะและจำนวนลูกค้า รวมถึงมีความต้องการไม่แน่นอน โดยผู้วิจัยตัดความไม่แน่นอนออกโดยใช้ค่าเฉลี่ยและฐานนิยม โดยสรุปการศึกษาวิจัยนี้ การปรับปรุงคำตอบด้วย Crossover-Move, 2-Opt และ One-Move ให้ผลระยะทางต่ำที่สุด โดยสามารถลดระยะทางจากเดิมได้ 24.46% มีข้อสังเกตเรื่องความต้องการลูกค้าเพราะเป็นพารามิเตอร์หนึ่งที่สำคัญ

ชุลีกร ชนะสิทธิ์; และสรวิชัย เยาวสุวรรณไชย (2556) พัฒนาโปรแกรมการจัดเส้นทางขนส่งสินค้าจากคลังสินค้า โดยใช้เทคนิค Nearest Neighbor Heuristics (NNH) ในการแก้ปัญหา TSP และ VRP ด้วยวิธี 2-Opt และ 3-Opt พบว่าการจัดเส้นทางขนส่งสินค้าแบบ

TSP-NNH สามารถลดระยะทางในการจัดเส้นทาง ได้ร้อยละ 11.79 และจากการปรับปรุงเส้นทางแบบ VRP-NNH ด้วยวิธี 2-Opt และ 3-Opt สามารถลดระยะทางได้ร้อยละ 6.13 และ 12.89 ตามลำดับ โดยสรุปการศึกษาวิจัยนี้ การพัฒนาโปรแกรมการจัดเส้นทางรถขนส่งสินค้าด้วยวิธีดังกล่าว สามารถช่วยเพิ่มประสิทธิภาพการจัดเส้นทางได้

ปาสีรัฐ บุญก่อน (2554) ศึกษาเกี่ยวกับวิธีฮิวริสติกส์แบบ Saving Algorithm ของ Clarke and Wright ด้วยวิธี 2-Opt ในการปรับปรุงเส้นทางเพื่อหาแนวทางแก้ปัญหาการจัดเส้นทางเดินพาหนะแบบมีข้อจำกัดเรื่องความสามารถของพาหนะในการบรรทุก โดยให้ได้ระยะทางสั้นที่สุด เทคนิค 2-Opt เป็นการปรับปรุงเส้นทางภายในเส้นทางเดียว ไม่มีการปรับปรุงระหว่างเส้นทาง วิธีการคือสลับ 2 ลำดับจุดส่งเพื่อเปรียบเทียบผลลัพธ์การจัดเส้นทางระหว่างก่อนปรับปรุงและหลังปรับปรุง เพื่อเลือกเส้นทางที่ให้ผลลัพธ์ที่ดีที่สุด โดยสรุปการศึกษาวิจัยนี้ สามารถเพิ่มประสิทธิภาพการขนส่งได้มากขึ้นด้วยเหตุเพราะระยะทางที่สั้นลง

ธัญชดา พันนิกุล; และสมบัติ สินธุชาว (Thanchuda Phannikul; and Sombat Sindhuchao. 2010) ศึกษาเกี่ยวกับวิธีแก้ปัญหาสำหรับปัญหา VRP ในกรณีที่มีการรับและส่งสินค้าในจุดเดียวกัน ซึ่งแบ่งเป็น 2 phase เริ่มจากการสร้างคำตอบเริ่มต้นโดยใช้วิธี Saving จากนั้นปรับปรุงคำตอบโดยใช้ Neighborhood Search (Insertion, Two-Opt, Crossover และ Reverse) โดยสรุปการศึกษาวิจัยนี้สามารถสร้างเส้นทางรถขนส่งได้ดี

โจส บรันดาว (Jose Brandao. 2007) ศึกษาการใช้ Tabu Search สำหรับกรณี Fleet Size and Mix Vehicle Routing Problem (FSMVRP) โดยมีจุดประสงค์เพื่อลดต้นทุนรวมให้ต่ำลง ลักษณะของปัญหานั้นประกอบด้วยความต้องการของลูกค้า การเข้าถึงลูกค้าเพียงหนึ่งครั้ง, ยานพาหนะเริ่มต้นและสิ้นสุดที่คลังพัสดุ และมีข้อจำกัดการบรรทุก ซึ่งหากไม่พิจารณาเรื่อง Fixed Costs จะมีลักษณะคล้ายกับ VRP ขั้นตอนการทำงานของโปรแกรมนั้นผู้วิจัยได้เริ่มจาก เลือกประเภทยานพาหนะ หลังจากนั้นจึงคำนวณไปหาลูกค้าที่ใกล้ที่สุด และวนซ้ำเรื่อยๆ จนของเต็มพาหนะ ขั้นตอนที่สองทำการเกลี่ยยานพาหนะให้แต่ละสายโดยอาศัยเทคนิค Insert จากนั้นขั้นตอนที่สามทำการกระบวน Giant Tour หลังจากทำการกระบวนการจัดสาย เข้าสู่ Tabu Search โดยใช้ Insert และ Swap มาประกอบการปรับปรุง โดยสรุป ผลการวิจัยพบว่าผลการคำนวณได้ผลดี โดยเปรียบเทียบกับผลงานวิจัยเก่าท่านอื่นๆ

แอสวิน โกเอล; และโวลเกอร์ กรุน (Asvin Goel; and Volker Gruhn. 2008) ศึกษาเกี่ยวกับความสำคัญของ Vehicle Routing Problem ที่รวมทั้งความหลากหลาย ข้อซ้อนเข้าด้วยกัน ซึ่งเรียกกลางเป็น General Vehicle Routing Problem (GVRP) ทั้งข้อจำกัดด้านการบรรทุก, ด้านเวลา, ชนิดรถ, ความต้องการของลูกค้า, ตำแหน่งเริ่มและสิ้นสุด, ตำแหน่งลูกค้าและข้อจำกัดด้านเส้นทาง ด้วยเหตุที่ GVRP มีข้อจำกัดหลากหลายด้าน อีกทั้งมีวิธีการแก้ปัญหานี้อยู่มากมาย ผู้วิจัยจึง

นำมาเพื่อศึกษาและพัฒนาเพิ่มเติมบนพื้นฐานแนวคิด Neighborhood Structure โดยมีแรงจูงใจธุรกิจ Air-Cargo ที่มีการเปลี่ยนแปลงหรือยกเลิกล่วงหน้าเป็นระยะเวลาดังนั้น Algorithm ที่ใช้กรณีที่มีการวางแผนมีการเปลี่ยนแปลงต้องใช้เวลาคำนวณที่เร็ว

Algorithm ที่ได้เป็นลักษณะ Reduced Variable Neighborhood Search (RVNS) ซึ่งศึกษาจาก Hansen และ Mladenovic (2003) โดย RVNS ประกอบด้วย 5 การดำเนินการ ดังนี้

INSERT สุ่มเลือกลูกค้าที่ไม่อยู่ในแผนตาราง แทรกเข้าเส้นทางรถโดยให้ต้นทุนที่เพิ่มขึ้นต่ำที่สุด

REMOVE สุ่มเลือกลูกค้าที่อยู่ในแผนตารางออกจากเส้นทางรถ

RELOCATE สุ่มเลือกลูกค้าที่อยู่ในแผนตารางออกจากเส้นทางรถ จากนั้นนำเข้าใหม่โดยให้ต้นทุนที่เพิ่มขึ้นต่ำที่สุด

REPLACE สุ่มเลือกลูกค้าที่อยู่ในแผนตารางออกจากเส้นทางรถ และแทนที่ด้วยอีกลูกค้าหนึ่ง

SWAP สุ่มเลือกลูกค้าที่อยู่ในแผนตาราง 2 จุด จากนั้นสลับที่

อีกหนึ่งส่วนคือผู้วิจัยนำเสนอการใช้ Large Neighborhood Search (LNS) โดย Algorithm นี้ เริ่มต้นจากตั้งเส้นทางแรก จากนั้นสร้างเส้นทางที่สองจากแทรกลูกค้าที่ไม่อยู่ในตารางแผนงาน จากนั้นเปรียบเทียบ หากเส้นทางใหม่ถูกยอมรับว่าดีกว่า ให้เปลี่ยนเป็นใช้เส้นทางใหม่ และดำเนินการจนครบจำนวนซ้ำ โดยสรุปการศึกษาวิจัยนี้ ค่าเฉลี่ยของเวลาการคำนวณของวิธีแรก ต่ำกว่าแบบที่สอง ซึ่งอาจจะเหมาะสมกับการแก้ปัญหากรณีไม่คงที่

กวี ศรีเมือง (2550) ศึกษาเกี่ยวกับวิธีการหาจำนวนพาหนะบรรทุกที่เหมาะสมในการขนส่งสินค้าในธุรกิจค้าปลีก Branch and Bound และ Saving Algorithm ของ Clark and Wright (1964) ในการแก้ปัญหาเส้นทางของสินค้าที่ต่อพ่วงกันไป หรือการขนส่งสินค้าแบบไม่เต็มคัน (LTL) เพื่อเพิ่มประสิทธิภาพของอัตราการบรรทุกในเชิงปริมาตร หรือเชิงน้ำหนัก อีกทั้งยังแสดงการจัดลำดับงานตามเวลาการส่งสินค้าและรอบการวิ่งของรถ โดยสรุปการศึกษาวิจัยนี้ สามารถประหยัดค่าขนส่งได้ 4.22% ต่อปี

ชัยยา นุรักษ์เช (2548) ศึกษาเกี่ยวกับเส้นทางขนส่งสินค้าจากศูนย์กระจายสินค้าไปลูกค้ารายย่อยในพื้นที่กรุงเทพมหานครและปริมณฑล โดยมีข้อจำกัดด้านเวลาคือ 8 ชั่วโมงในการขนส่งแต่ละเที่ยว ผู้วิจัยได้นำกรณีศึกษาจากทางโอ. ซี. ซัพพลาย และใช้วิธีการทางฮิวริสติกส์ในการจัดเส้นทาง ทั้งนี้เพื่อให้ได้เส้นทางขนส่งที่มีระยะทางรวมต่ำที่สุด วิธีการทางฮิวริสติกส์ที่ผู้วิจัยนำมาใช้คือ Nearest Insertion Approach และ 2-Opt โดยสรุปการศึกษาวิจัยนี้สามารถให้ระยะทางได้สั้นลง

คลอดีโอ บี; และมาร์คอส โรเบอร์โต ซิลวา (Claudio B; and Marcos Roberto Silva. 2005) ศึกษาการใช้ Genetic Algorithm สำหรับการออกแบบการส่งสินค้าแบบ Hub and Spoke ด้วยวิธีการบรรทุกสินค้าแบบ Less than Truckload (LTL) เพื่อให้ต้นทุนต่ำที่สุด (ประกอบด้วย ต้นทุนโดยตรงและต้นทุนผันแปร) ด้วยเหตุผลที่ปัจจุบันในประเทศบราซิลมีการขนส่งแบบ Less than Truckload เป็นส่วนใหญ่ ผู้วิจัยจึงได้นำปัญหานี้มาทำการศึกษา โดยใช้ Genetic Algorithm Heuristic สำหรับหาจำนวนและที่ตั้งของ Hubs รวมถึงเส้นทางจาก Spoke ไป Hubs

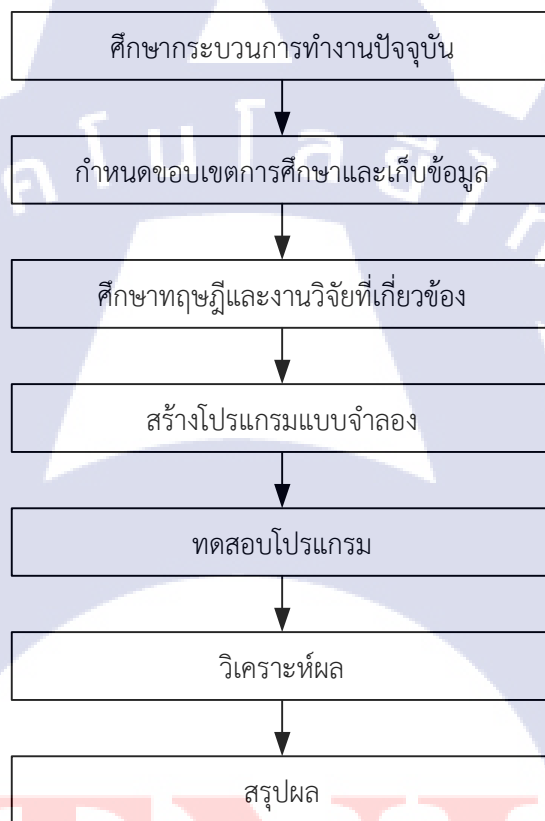
เริ่มจากสร้างกลุ่มเริ่มต้นโดยการสุ่ม จากนั้นเลือก 2 วิธีสำหรับเป็นตัวตั้งต้นจากนั้นเลือก เพียง 1 คำตอบที่ได้ ที่เป็นคำตอบที่ดีที่สุดตามจุดประสงค์ที่ต้องการ มาปรับปรุงต่อด้วยการ Cross Over ทำเรื่อยๆ ตามจำนวนรอบที่กำหนด โดยสรุปงานวิจัยพบว่าการดำเนินการปรับปรุงด้วย GA ได้ผลดี สามารถช่วยการดำเนินการของธุรกิจที่ศึกษานี้ได้



TNI

บทที่ 3 ระเบียบวิธีวิจัย

การดำเนินงานวิจัยจะเริ่มจากศึกษากระบวนการทำงานปัจจุบัน โดยมีขั้นตอนต่างๆ ดังรูปที่ 3.1



รูปที่ 3.1 แผนภาพแสดงกระบวนการทำวิจัย

3.1 ศึกษากระบวนการปัจจุบัน

3.1.1 การคำนวณเส้นทางรถที่ใช้ในปัจจุบัน

อ้างอิงจาก Master Route ที่ตั้งไว้ในตอนเริ่มต้นเปิดใช้ศูนย์กระจายสินค้า แต่ในแต่ละวันนั้น มียอดการสั่งสินค้าจากร้านมาที่ศูนย์กระจายสินค้าไม่คงที่ อีกทั้งกรณีที่ร้านสาขาเปิดใหม่ในวันนั้นๆ ผู้ปฏิบัติหน้างานจะดำเนินการจัดร้านเพิ่มเข้าไปในแต่ละเส้นทางรถ ด้วยทักษะ ความชำนาญ และประสบการณ์ของพนักงานเอง เนื่องจากร้านที่เพิ่มขึ้นมานั้น ไม่มีในเส้นทางรถสายหลักที่

ประมวลผลมาจากโปรแกรมสำเร็จรูป ซึ่งนั่นเป็นข้อจำกัดบางประการของโปรแกรมสำเร็จรูปนี้ ทำให้ไม่สามารถนำมาใช้วางแผนเส้นทางพาหนะขนส่งประจำวันได้

3.1.2 ศึกษาและเก็บรวบรวมปัญหา

จากโปรแกรมสำเร็จรูปที่ใช้สำหรับทำ Master Route ซึ่งพบว่ามีปัญหาเรื่องของการดำเนินการ โดยใช้เวลาในการดำเนินการจัดเส้นทางรถแต่ละครั้งไม่ต่ำกว่า 4 ชั่วโมง และไม่สามารถคำนวณแยกประเภทรถได้

3.1.3 ศึกษาโปรแกรมสำเร็จรูปอื่นๆ

จากการศึกษาโปรแกรมสำเร็จรูปอื่นๆ ที่เกี่ยวกับ Transportation Management พบว่าโปรแกรมสำเร็จรูปที่มีอยู่ สามารถรองรับการทำงานได้ แต่ต้องมีค่าใช้จ่ายเพิ่ม และยังไม่สามารถประเมินเวลาการทำงานของโปรแกรมในกรณีศึกษานี้ได้ อีกทั้ง ความสามารถของโปรแกรมอื่นๆ เกินขอบเขตที่ต้องการในขณะนี้ (เนื่องจากการใช้งานอื่นๆ ทางบริษัทมีระบบรองรับอยู่ในปัจจุบัน)

3.2 กำหนดขอบเขตการศึกษา และเก็บข้อมูล

กำหนดขอบเขตการศึกษาและเก็บรวบรวมปัจจัยที่เกี่ยวข้อง ในการปรับเส้นทางรถแต่ละวัน พบว่ามีข้อมูลที่ใช้ดำเนินการ ดังนี้

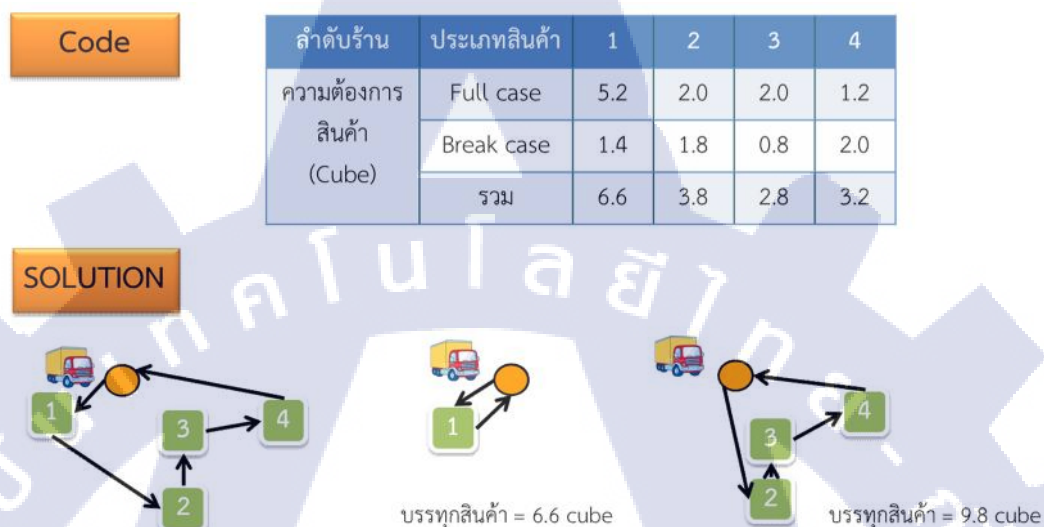
- ยอดส่งสินค้าในวันนั้น รายร้าน (หน่วย : Cube) แยกประเภทสินค้า Full Case และ Break Case
- ระยะทางจากทุกตำแหน่งร้านและศูนย์กระจายสินค้า
- ประเภทรถ จำนวนรถ และ ความจุของรถแต่ละประเภท
- ราคาค่าขนส่ง

3.3 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง

จากลักษณะปัญหาที่ได้กล่าว เมื่อนำมาศึกษาร่วมกับทฤษฎี พบว่าเป็นรูปแบบปัญหา VRP ประเภท CVRP เนื่องจากการพิจารณาเรื่องความจุของรถ และจากการศึกษาวิธีการปัญหาการจัดเส้นทางสำหรับยานพาหนะ ได้นำวิธี Variable Neighborhood Search โดยระบุการปรับปรุงเส้นทางโดยใช้ SWAP, INSERT และ REVERSE มาช่วยในการจัดเส้นทาง รวมไปถึงจัดการบรรทุกสินค้าอีก 2 รูปแบบ เพื่อเปรียบเทียบลักษณะการขนส่งสินค้า ที่มีผลกับจำนวนรถที่ใช้ รูปแบบการบรรทุกทุกสินค้า ดำเนินการ 2 รูปแบบ ดังนี้

3.3.1 บรรทุกสินค้าไม่เต็มคันรถ (Less Than Truckload)

ให้รถบรรทุกสินค้า ตามลำดับร้าน เมื่อไปถึงร้านได้แล้ว บรรทุกอีกไม่ได้ ให้บรรทุกร้านนั้นเป็นร้านสุดท้าย

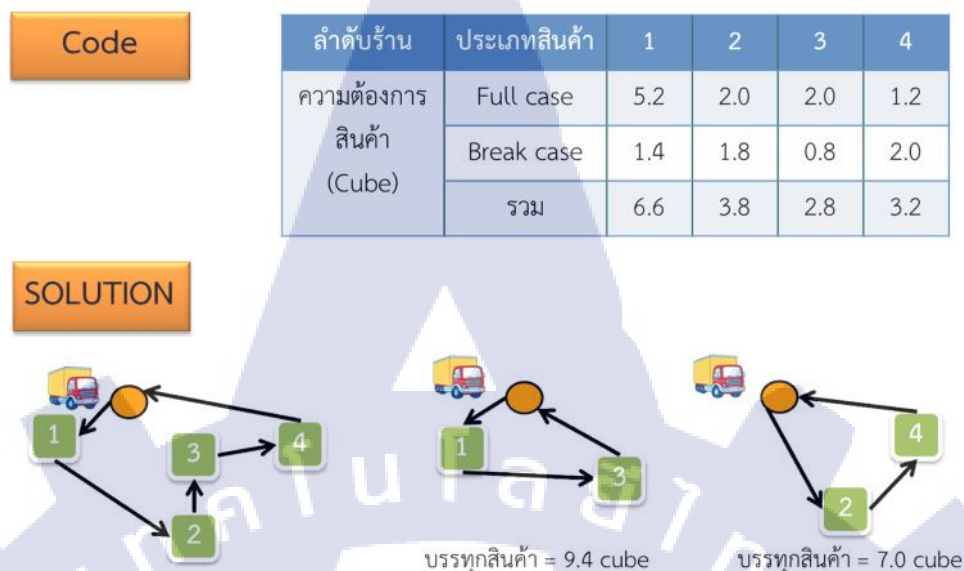


รูปที่ 3.2 รูปแบบการคำนวณการใช้รถแบบ Less than Truckload

3.3.2 บรรทุกสินค้าเต็มคันรถ (Full Truckload)

ให้รถบรรทุกสินค้า ตามลำดับร้าน และสามารถข้ามไปร้านถัดไปได้ หากยังสามารถบรรทุกได้อยู่ โดยจำกัดขอบเขตระยะทางไม่เกิน 50 กม. จากร้านก่อนหน้า

TNI



รูปที่ 3.3 รูปแบบการคำนวณการใช้รถแบบ Full Truckload

3.4 สร้างโปรแกรมแบบจำลอง และทดสอบโปรแกรม

ในงานวิจัยนี้ ดำเนินการสร้างการคำนวณผ่านโปรแกรม Microsoft Visual Studio Express for Desktop โดยนำเข้าข้อมูลระยะทาง ยอดสั่งสินค้าของร้าน และราคาค่าขนส่ง ซึ่งเป็น Excel File จากนั้นผลลัพธ์ที่แสดง คือ ลำดับร้าน ระยะทาง ค่าขนส่ง และจำนวนรถแต่ละประเภท

3.4.1 เครื่องมือที่ใช้

- ฮาร์ดแวร์ : โน้ตบุ๊ก (Windows 8, 1.75 GHz AMD E2-2000 APU Processor)
- ซอฟต์แวร์ : Microsoft Visual Studio Express for Desktop, Microsoft

Excel

3.4.2 การวิเคราะห์

เปรียบเทียบค่าเฉลี่ยจากการคำนวณทั้ง 16 วิธี กับการคำนวณแบบปัจจุบัน ในด้าน ระยะทางและค่าใช้จ่าย ซึ่งสรุปการคำนวณได้ดังนี้

ตารางที่ 3.1 รูปแบบการคำนวณ

วิธี	ปรับปรุงเส้นทาง	คำนวณการใช้รถ
1	SWAP	LTL
2	SWAP	FTL
3	INSERT	LTL
4	INSERT	FTL
5	REVERSE	LTL
6	REVERSE	FTL
7	SWAP-INSERT	LTL
8	SWAP-INSERT	FTL
9	INSERT-SWAP	LTL
10	INSERT-SWAP	FTL
11	SWAP-REVERSE	LTL
12	SWAP-REVERSE	FTL
13	REVERSE-SWAP	LTL
14	REVERSE-SWAP	FTL
15	INSERT-REVERSE	LTL
16	INSERT-REVERSE	FTL

3.4.3 ข้อมูลที่ใช้ในการคำนวณ

ตารางที่ 3.2 ข้อมูลและรูปแบบข้อมูลที่ใช้

ข้อมูล	รูปแบบข้อมูล
ปริมาณการสั่งสินค้าของร้าน	ยอดสั่งสินค้า เฉลี่ยต่อวัน (แบ่งเป็น 3 ระดับ) - ยอดสั่งสินค้าสูง - ยอดสั่งสินค้าปานกลาง - ยอดสั่งสินค้าต่ำ
จำนวนรถและขนาดบรรจุ	แยกตามประเภทรถ สี่ล้อ และหกล้อ
ระยะทาง	ตาราง Metric ระยะทาง
ค่าขนส่ง	แยกตามประเภทรถ สี่ล้อ และหกล้อ แยกตามโซนของตำแหน่งร้าน

3.4.3.1 ปริมาณการสั่งสินค้าของร้าน (ข้อมูลเฉลี่ยต่อวัน)

ตารางที่ 3.3 ค่าเฉลี่ยยอดสั่งสินค้า ต่อร้าน ต่อวัน

ยอดสั่งสินค้าเฉลี่ยต่อวัน	ค่าเฉลี่ยยอดสั่ง ต่อร้าน ต่อวัน (หน่วยคิว)		
	Full Case	Break Case	Total
ช่วงยอดสั่ง สูง	5.4	1.4	6.8
ช่วงยอดสั่ง ปานกลาง	3.8	1.4	5.2
ช่วงยอดสั่ง ต่ำ	2.0	2.9	4.9

3.4.3.2 จำนวนรถและขนาดบรรจุ

ตารางที่ 3.4 จำนวนรถและขนาดบรรจุ

ประเภทรถ	มาตรฐานความจุ (Cube)	จำนวนรถ (คัน)
6 ล้อ	10	80
4 ล้อ	5.5	65

3.4.3.3 ข้อมูลระยะทาง

จัดทำในรูปแบบตารางเมตริกซ์ ตามจำนวนร้านกลุ่มศึกษา

ตารางที่ 3.5 รูปแบบข้อมูลระยะทาง

ชื่อร้าน	ระยะทาง										
	DCKK	A	B	C	D	E	F	G	H	I	J
DCKK	0	183.25	182.83	268.97	228.89	185.11	120.72	183.12	95.8	192.93	244.17
A	183.25	0	1.72	101.77	54.49	3	115.2	0.64	232.7	9.99	218.71
B	182.83	1.4	0	103.01	55.84	4.48	115.98	1.07	231.57	10.13	219.41
C	268.97	86.12	103.01	0	51.96	99.24	167.91	102.22	318.73	95.83	225.01
D	228.89	46.02	55.84	51.96	0	51.7	135.14	54.97	276.71	47.97	215.5
E	185.11	2.2	4.48	99.24	51.7	0	115.01	3.54	234.98	8.36	218.54
F	120.72	118.52	115.98	167.91	135.14	115.01	0	115.51	207.03	121.24	138.36
G	183.12	0.49	1.07	102.22	54.97	3.54	115.51	0	232.3	10.02	218.99
H	95.8	233.7	231.57	318.73	276.71	234.98	207.03	232.3	0	238.09	308.05
I	192.93	10.04	10.13	95.83	47.97	8.36	121.24	10.02	238.09	0	224.17
J	244.17	270.37	219.41	225.01	215.5	218.54	138.36	218.99	308.05	224.17	0

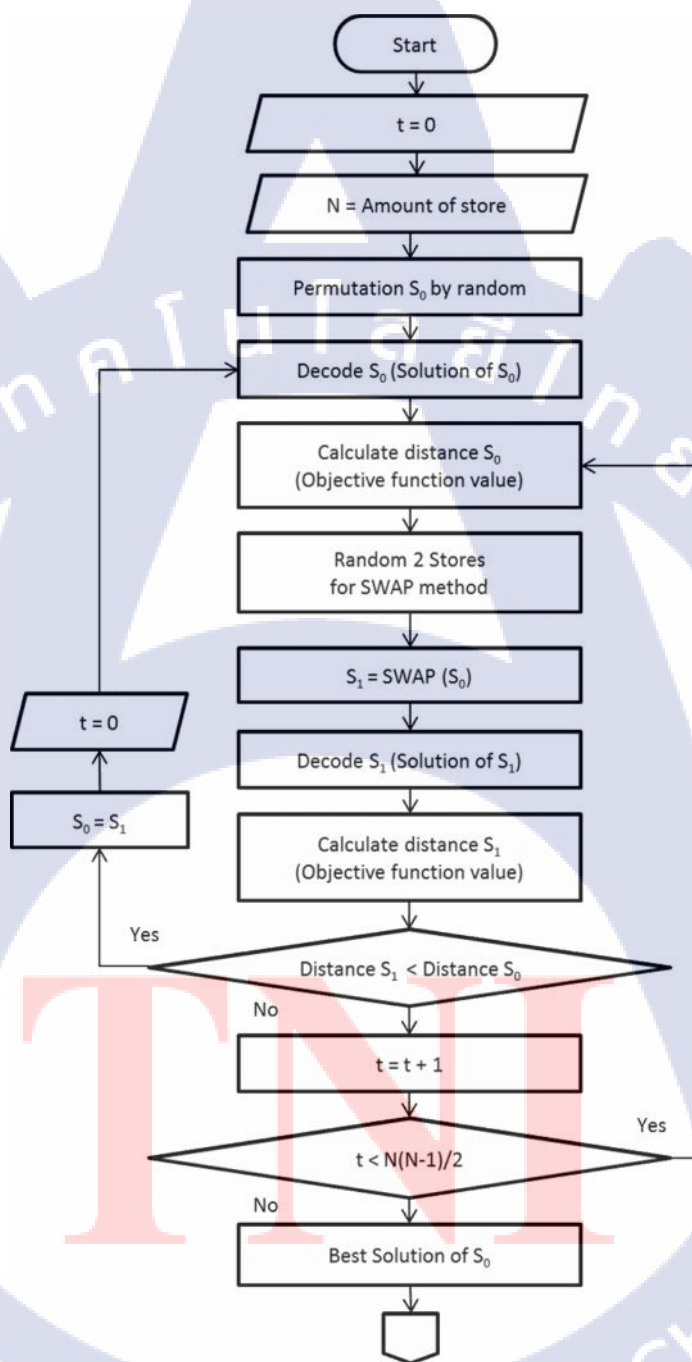
3.4.3.4 ค่าขนส่ง

ตารางที่ 3.6 รูปแบบข้อมูลค่าขนส่ง

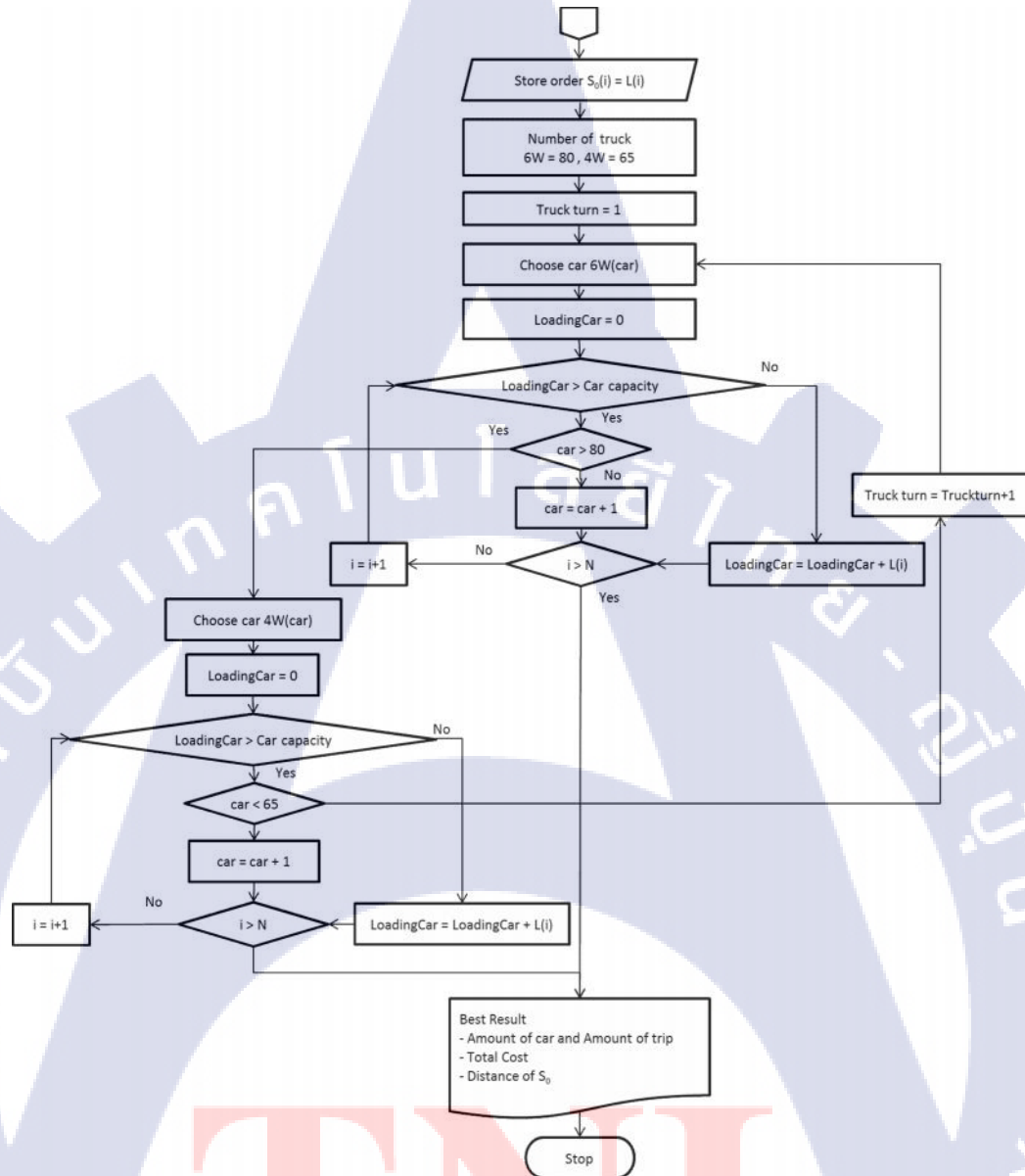
ร้าน	ค่าขนส่ง 4 ล้อ	ค่าขนส่ง 6 ล้อ	โซน
A	504	980	XX1
B	504	980	XX1
C	734	1236	XX7
D	634	1123	XX6
E	504	980	XX1
F	425	822	XX5
G	504	980	XX1
H	415	776	XX4
I	504	980	XX1
J	584	1300	XX2
K	474	1047	XX3

3.4.4 Flowchart

3.4.4.1 การปรับปรุงเส้นทางแบบ SWAP และคำนวณการใช้รถแบบ LTL

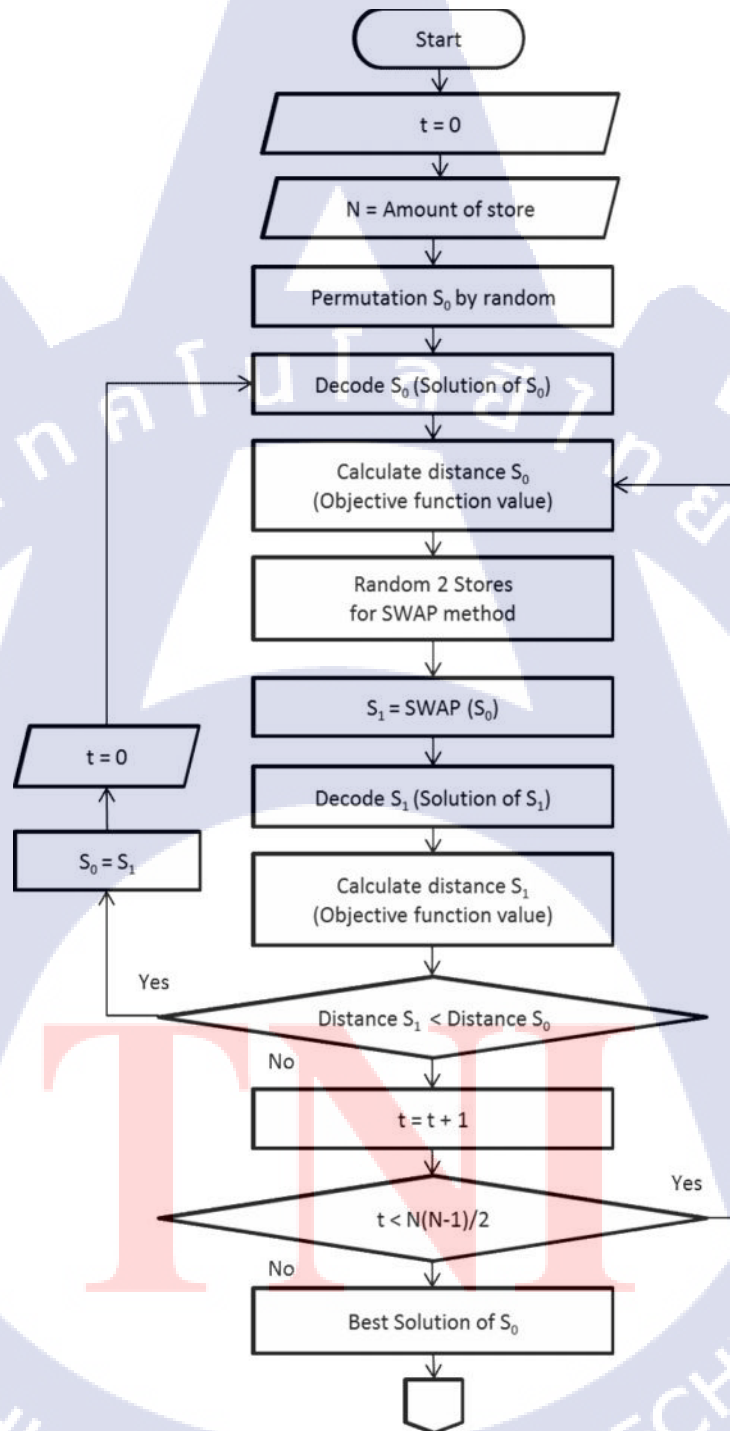


รูปที่ 3.4 หลักการการปรับปรุงเส้นทางแบบ SWAP และคำนวณการใช้รถแบบ LTL

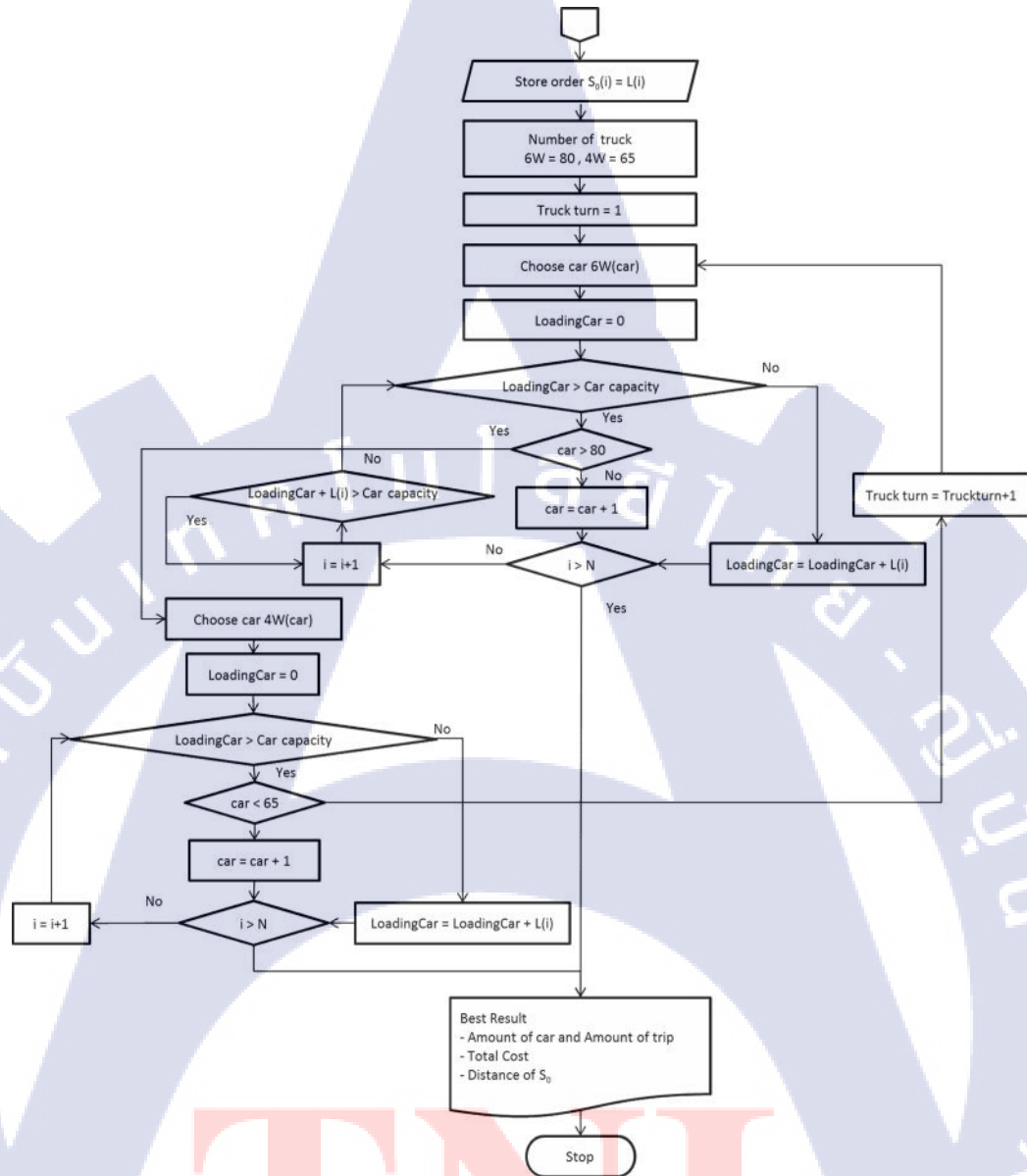


รูปที่ 3.4 หลักการการปรับปรุงเส้นทางแบบ SWAP และคำนวณการใช้รถแบบ LTL (ต่อ)

3.4.4.2 การปรับปรุงเส้นทางแบบ SWAP และคำนวณการใช้รถแบบ FTL

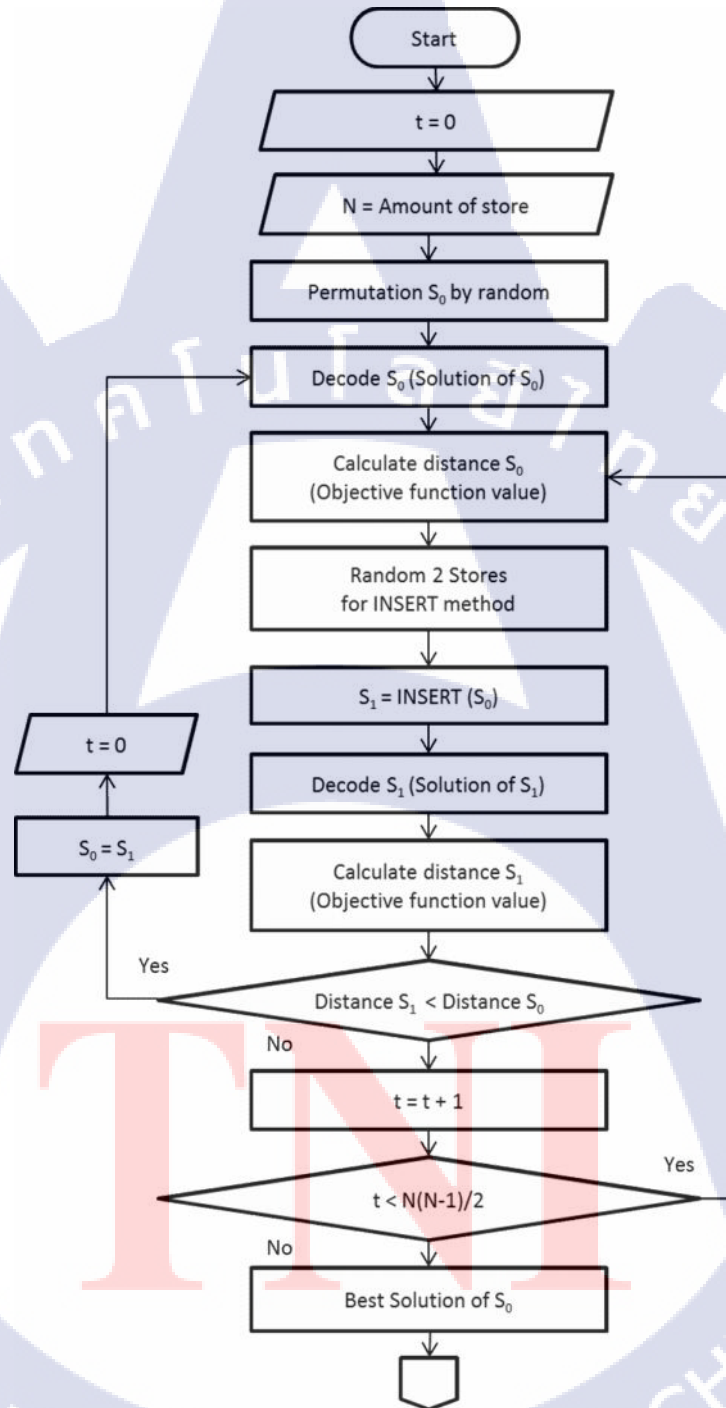


รูปที่ 3.5 หลักการการปรับปรุงเส้นทางแบบ SWAP และคำนวณการใช้รถแบบ FTL

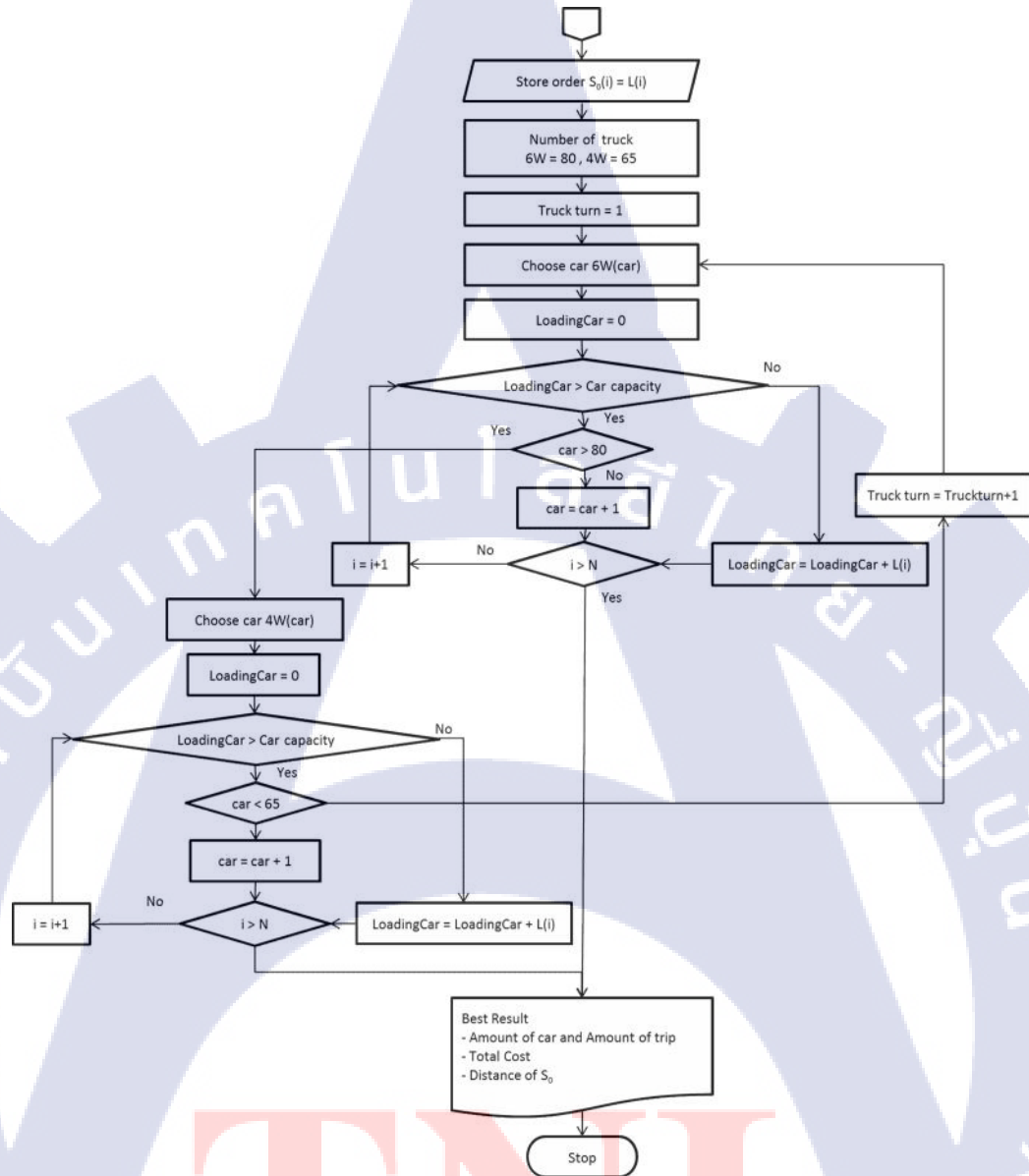


รูปที่ 3.5 หลักการการปรับปรุงเส้นทางแบบ SWAP และคำนวณการใช้รถแบบ FTL (ต่อ)

3.4.4.3 การปรับปรุงเส้นทางแบบ INSERT และคำนวณการใช้รถแบบ LTL

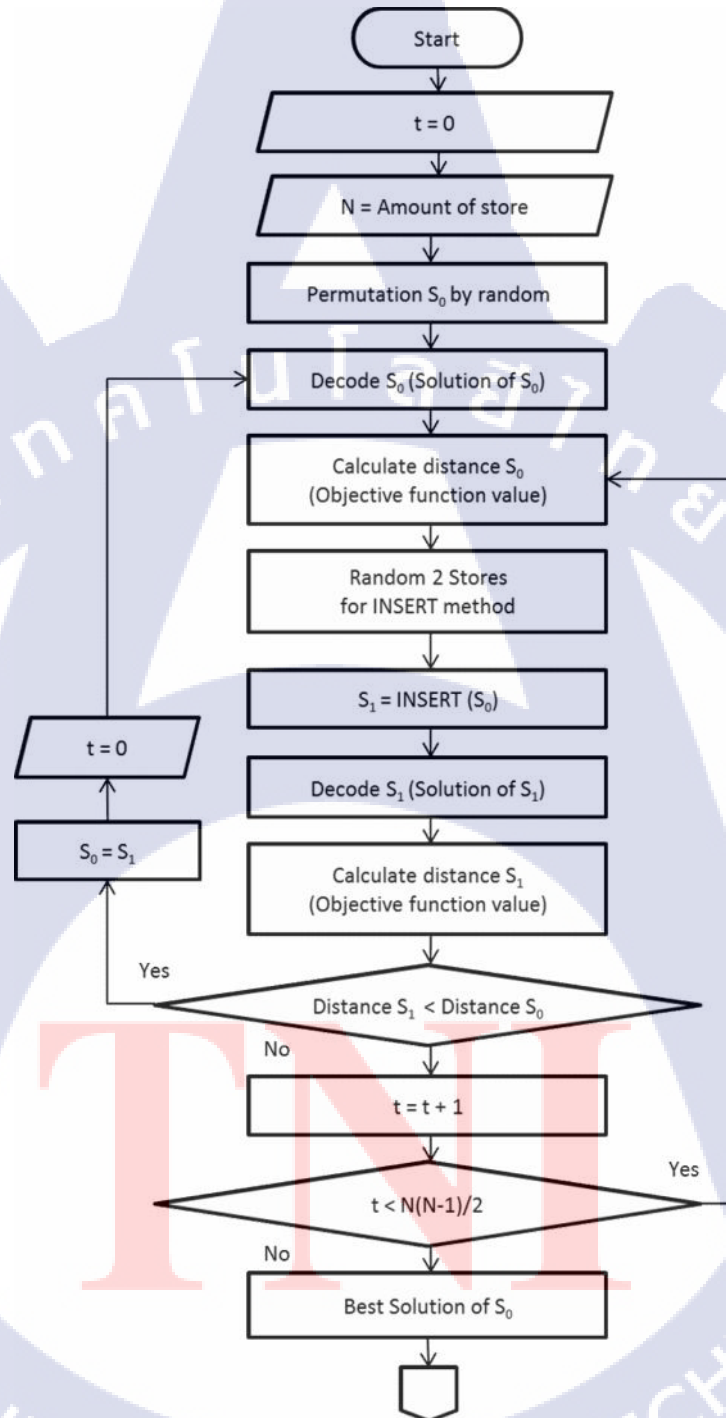


รูปที่ 3.6 หลักการการปรับปรุงเส้นทางแบบ INSERT และคำนวณการใช้รถแบบ LTL

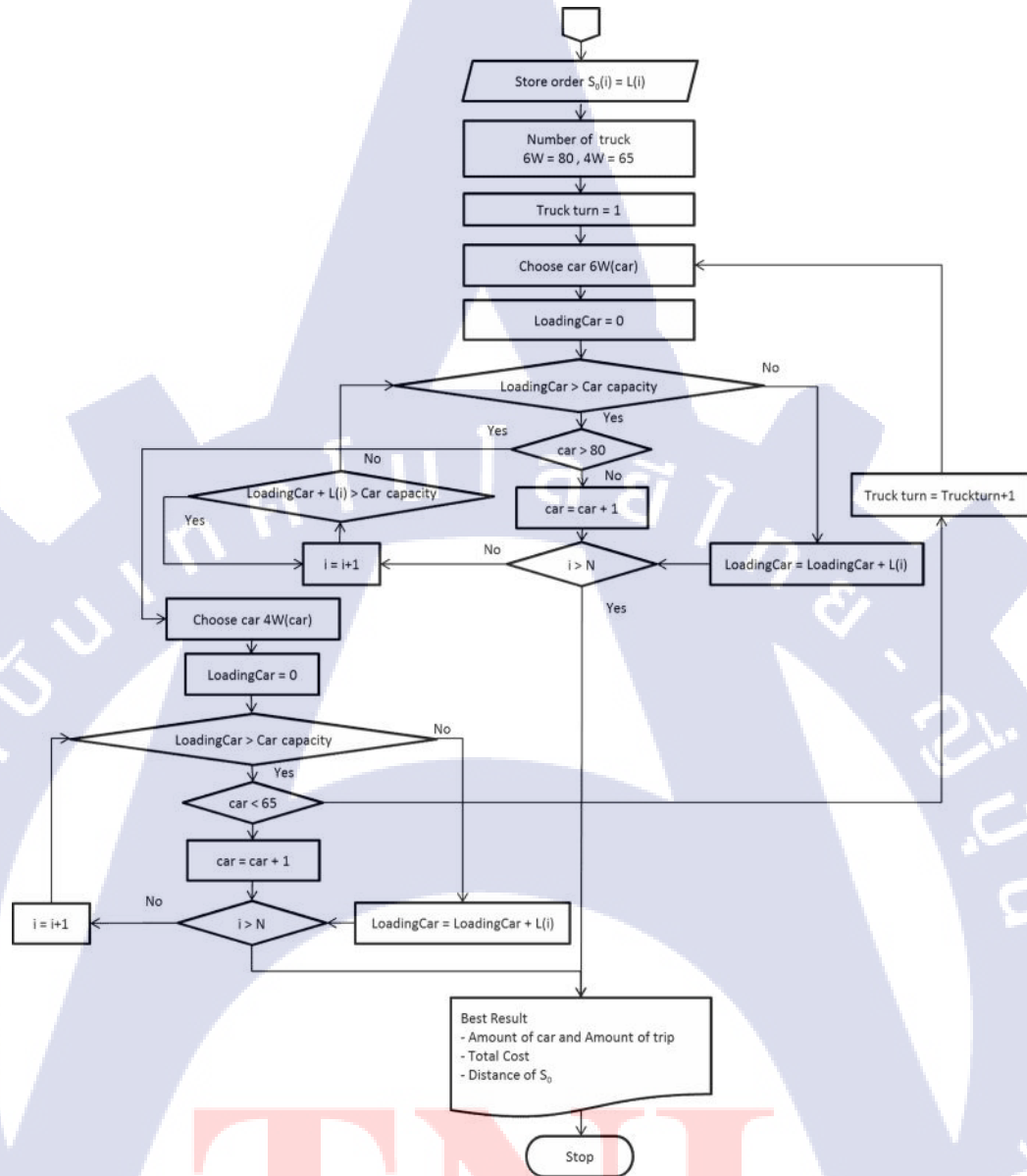


รูปที่ 3.6 หลักการการปรับปรุงเส้นทางแบบ INSERT และคำนวณการใช้รถแบบ LTL (ต่อ)

3.4.4.4 การปรับปรุงเส้นทางแบบ INSERT และคำนวณการใช้รถแบบ FTL

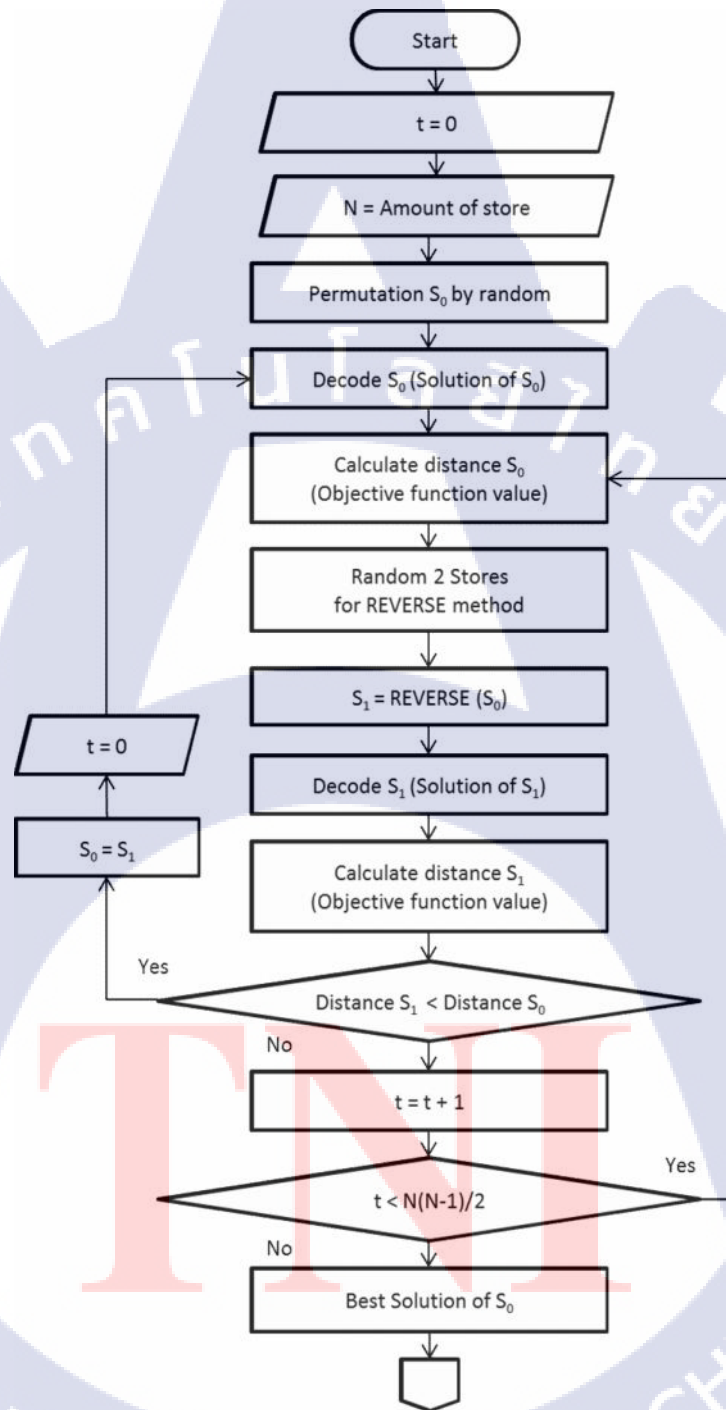


รูปที่ 3.7 หลักการการปรับปรุงเส้นทางแบบ INSERT และคำนวณการใช้รถแบบ FTL

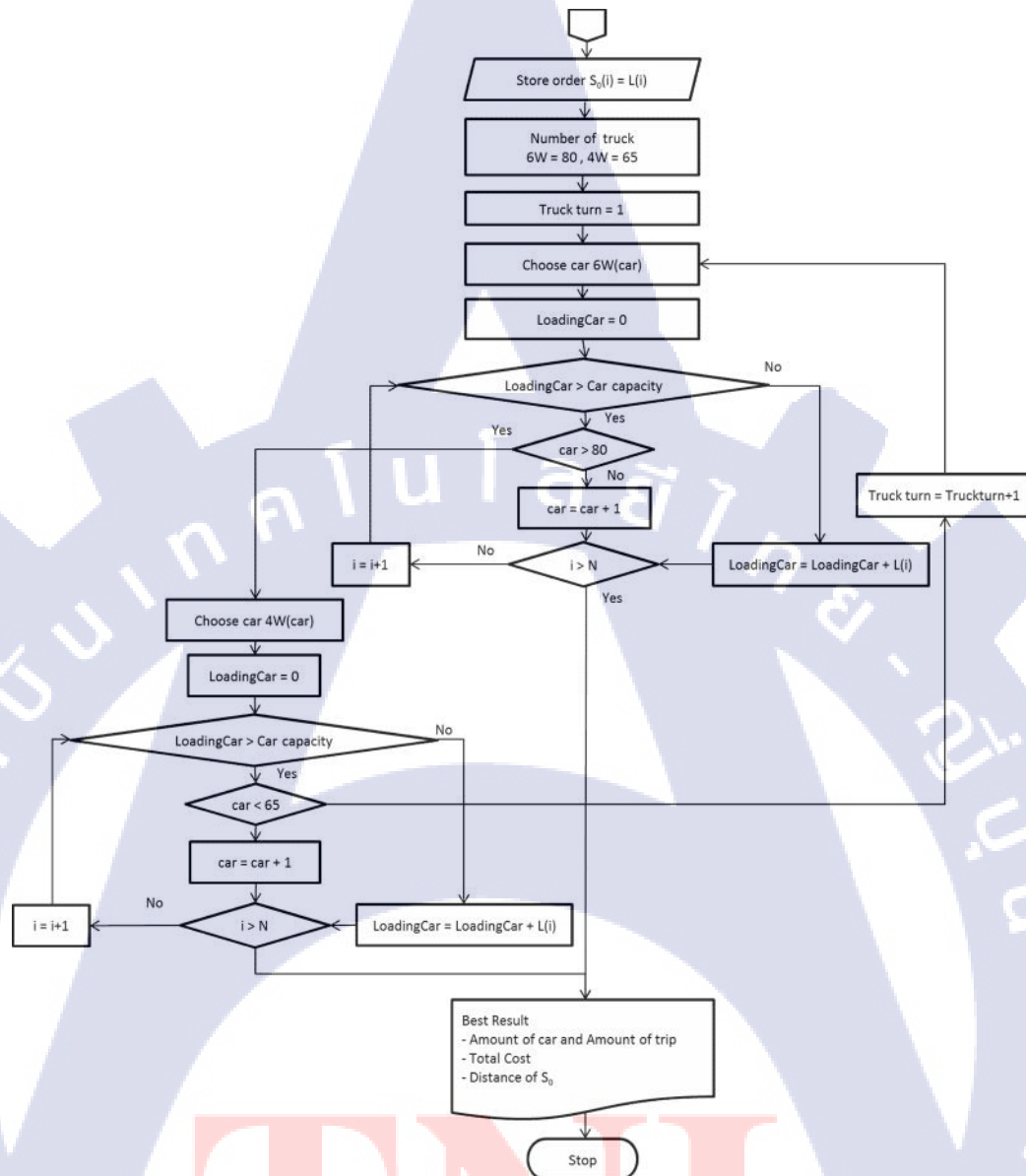


รูปที่ 3.7 หลักการการปรับปรุงเส้นทางแบบ INSERT และคำนวณการใช้รถแบบ FTL (ต่อ)

3.4.4.5 การปรับปรุงเส้นทางแบบ REVERSE และคำนวณการใช้รถแบบ LTL

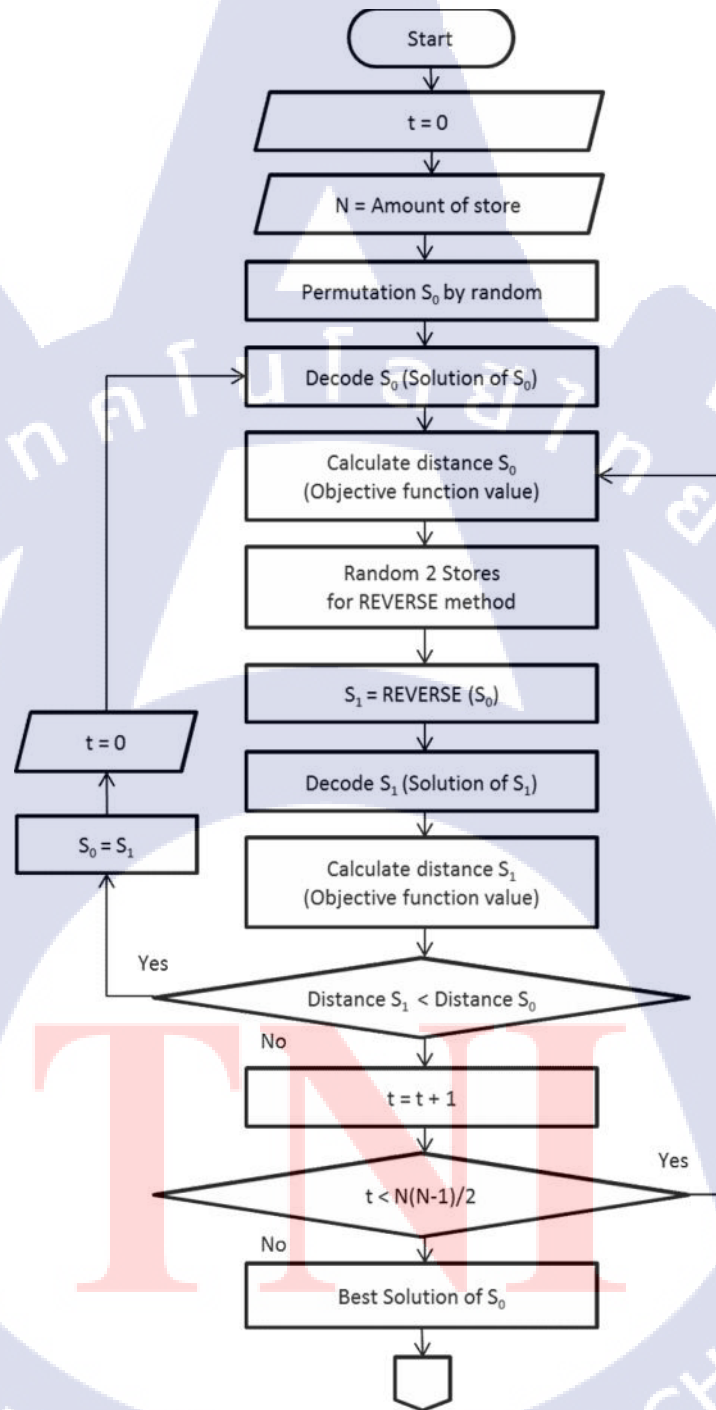


รูปที่ 3.8 หลักการการปรับปรุงเส้นทางแบบ REVERSE และคำนวณการใช้รถแบบ LTL

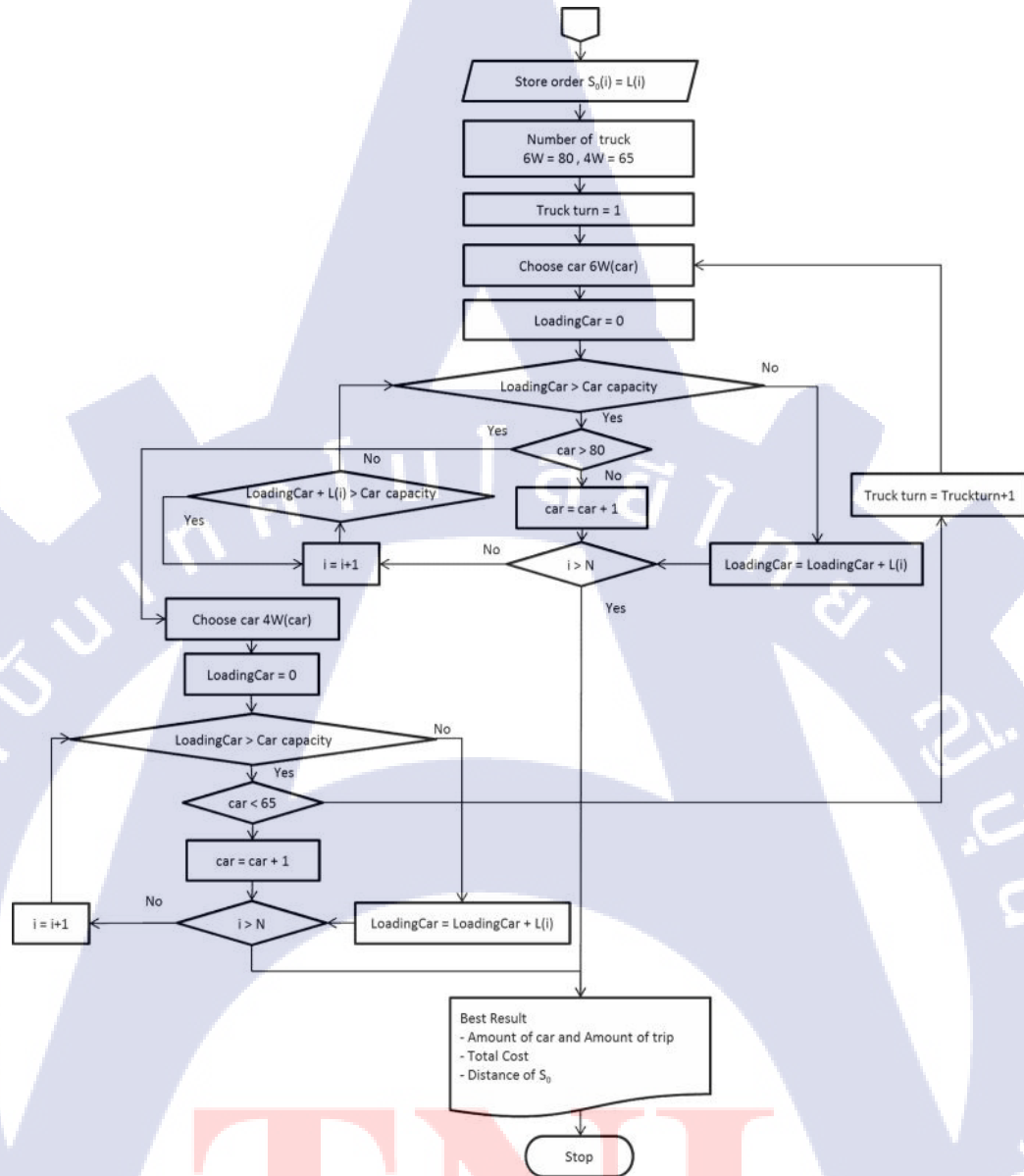


รูปที่ 3.8 หลักการการปรับปรุงเส้นทางแบบ REVERSE และคำนวณการใช้รถแบบ LTL (ต่อ)

3.4.4.6 การปรับปรุงเส้นทางแบบ REVERSE และคำนวณการใช้รถแบบ FTL

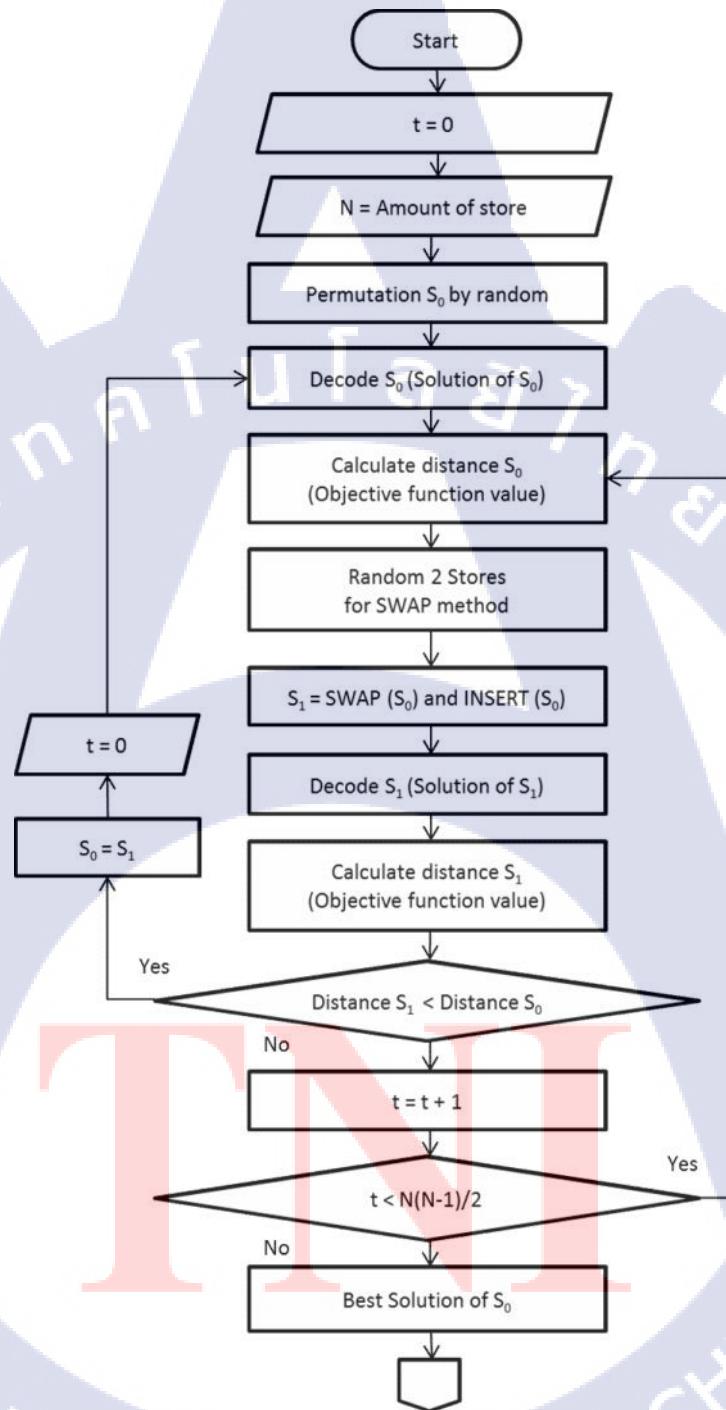


รูปที่ 3.9 หลักการการปรับปรุงเส้นทางแบบ REVERSE และคำนวณการใช้รถแบบ FTL

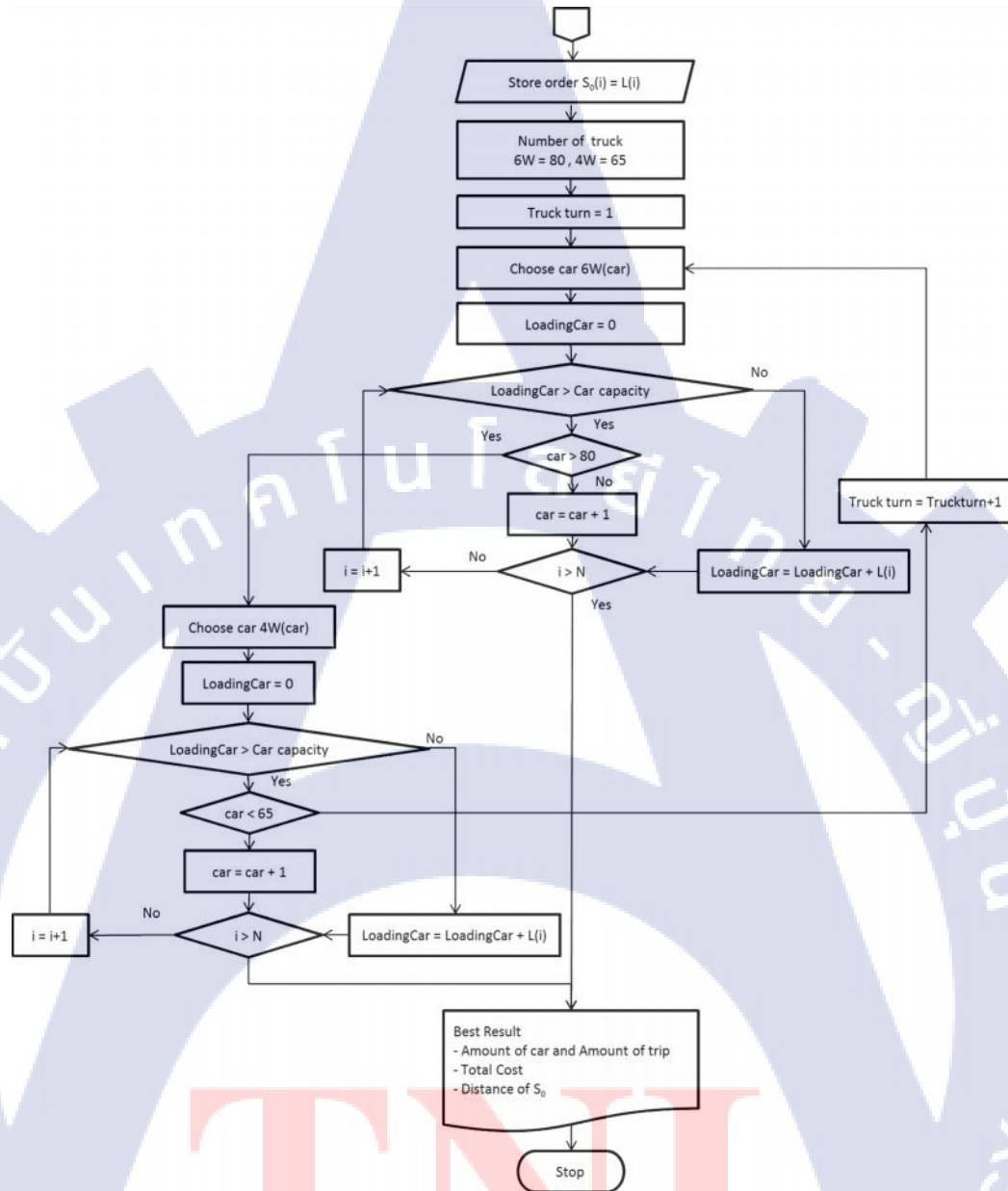


รูปที่ 3.9 หลักการการปรับปรุงเส้นทางแบบ REVERSE และคำนวณการใช้รถแบบ FTL (ต่อ)

3.4.4.7 การปรับปรุงเส้นทางแบบ SWAP-INSERT และคำนวณการใช้รถแบบ LTL

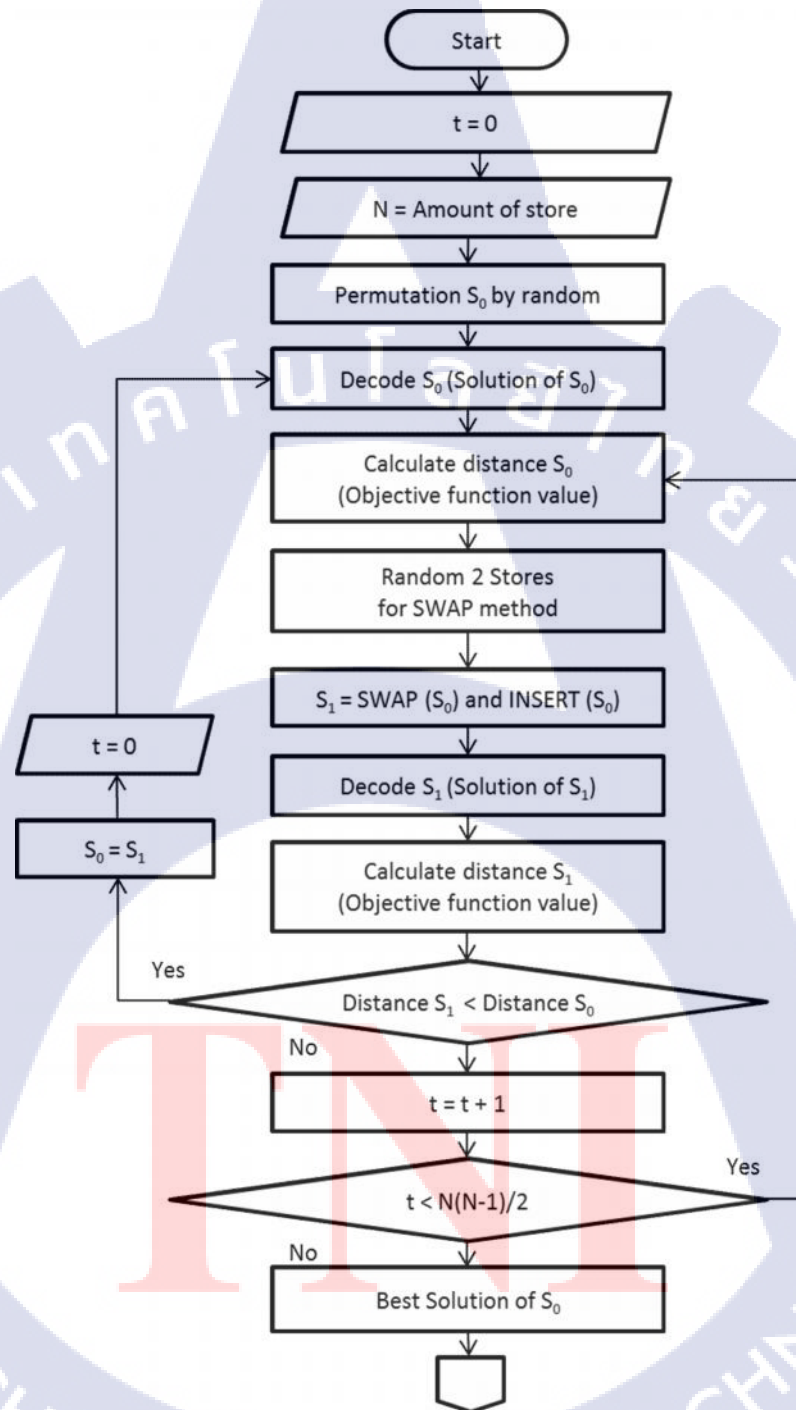


รูปที่ 3.10 หลักการการปรับปรุงเส้นทางแบบ SWAP-INSERT และคำนวณการใช้รถแบบ LTL

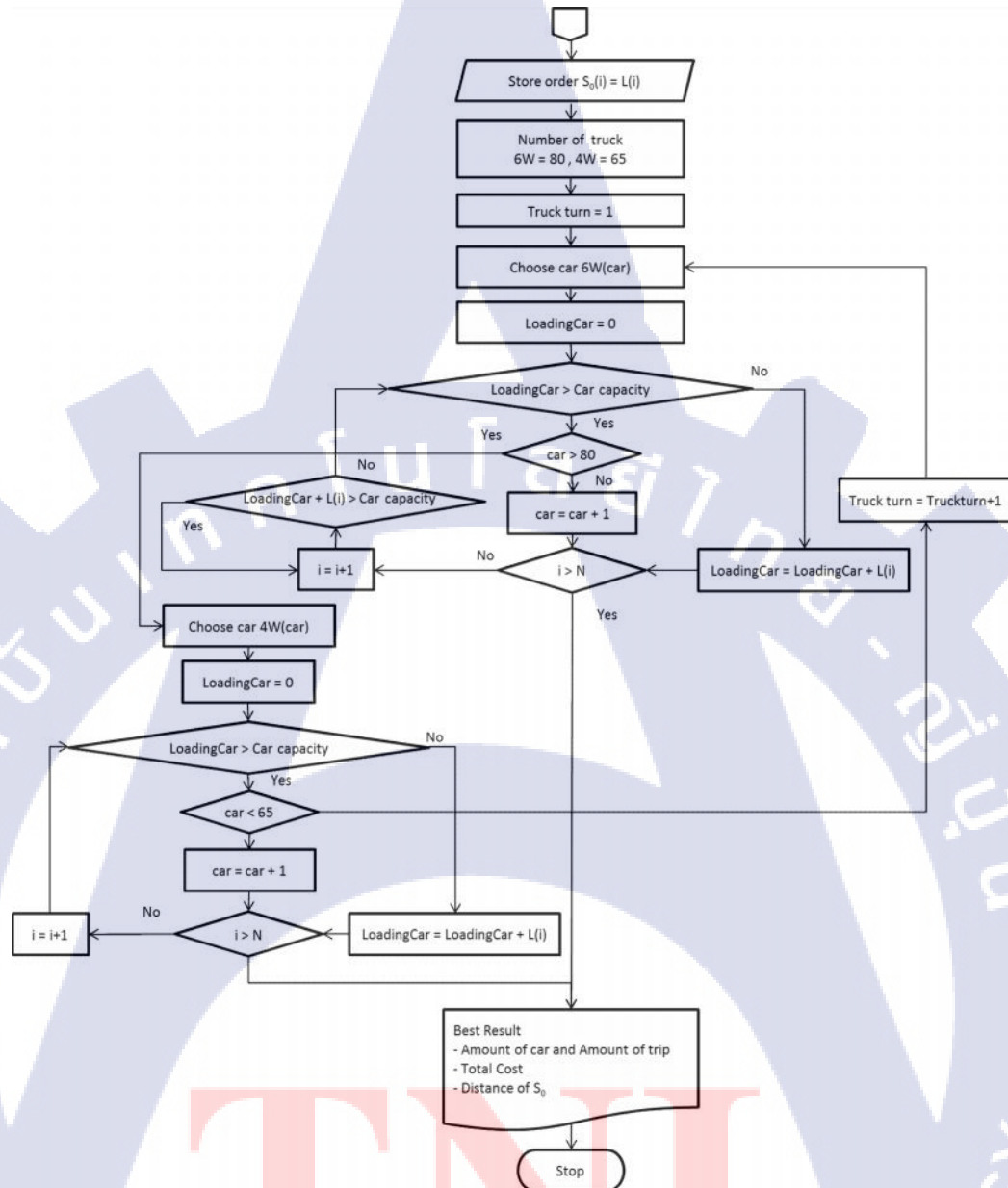


รูปที่ 3.10 หลักการการปรับปรุงเส้นทางแบบ SWAP-INSERT และคำนวณการใช้รถแบบ LTL (ต่อ)

3.4.4.8 การปรับปรุงเส้นทางแบบ SWAP-INSERT และคำนวณการใช้รถแบบ FTL

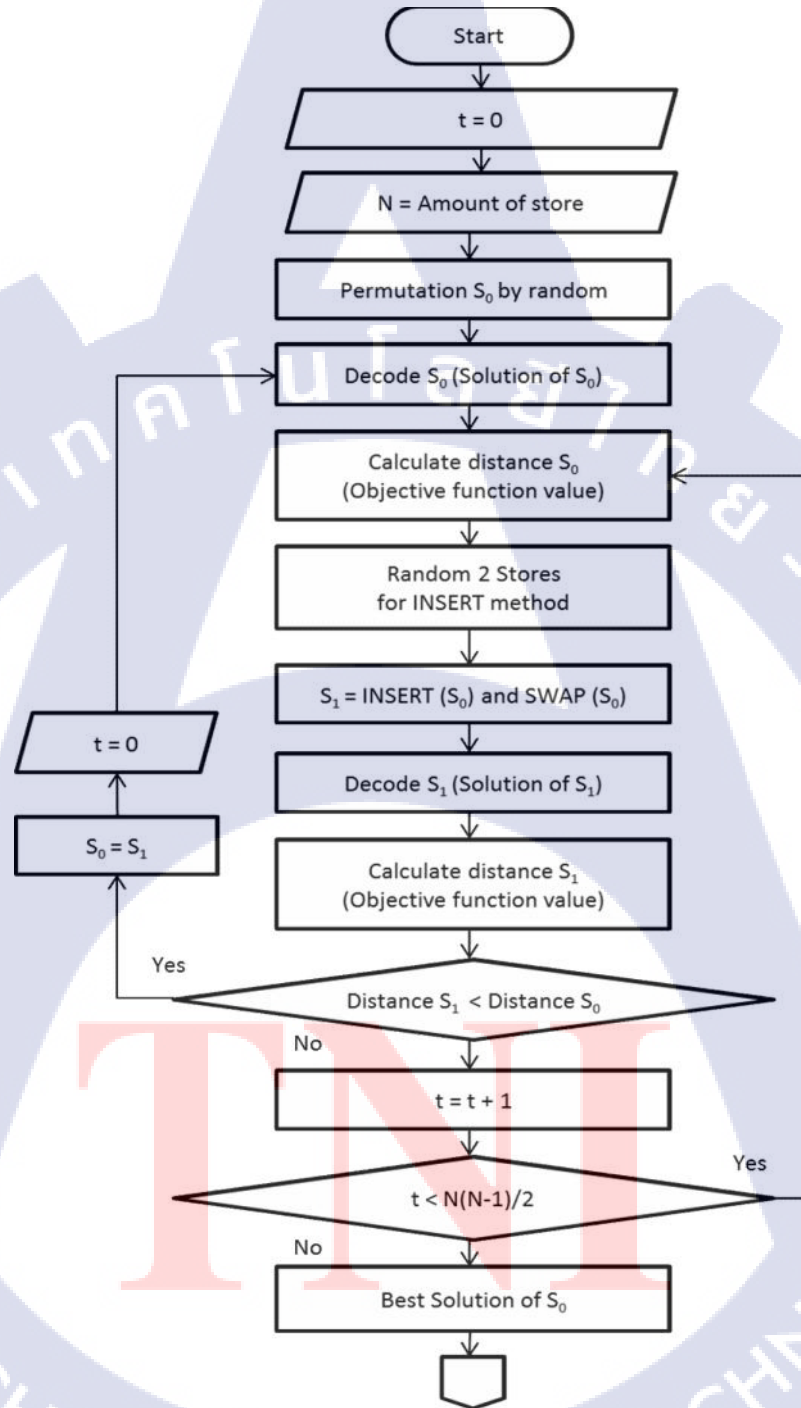


รูปที่ 3.11 หลักการการปรับปรุงเส้นทางแบบ SWAP-INSERT และคำนวณการใช้รถแบบ FTL

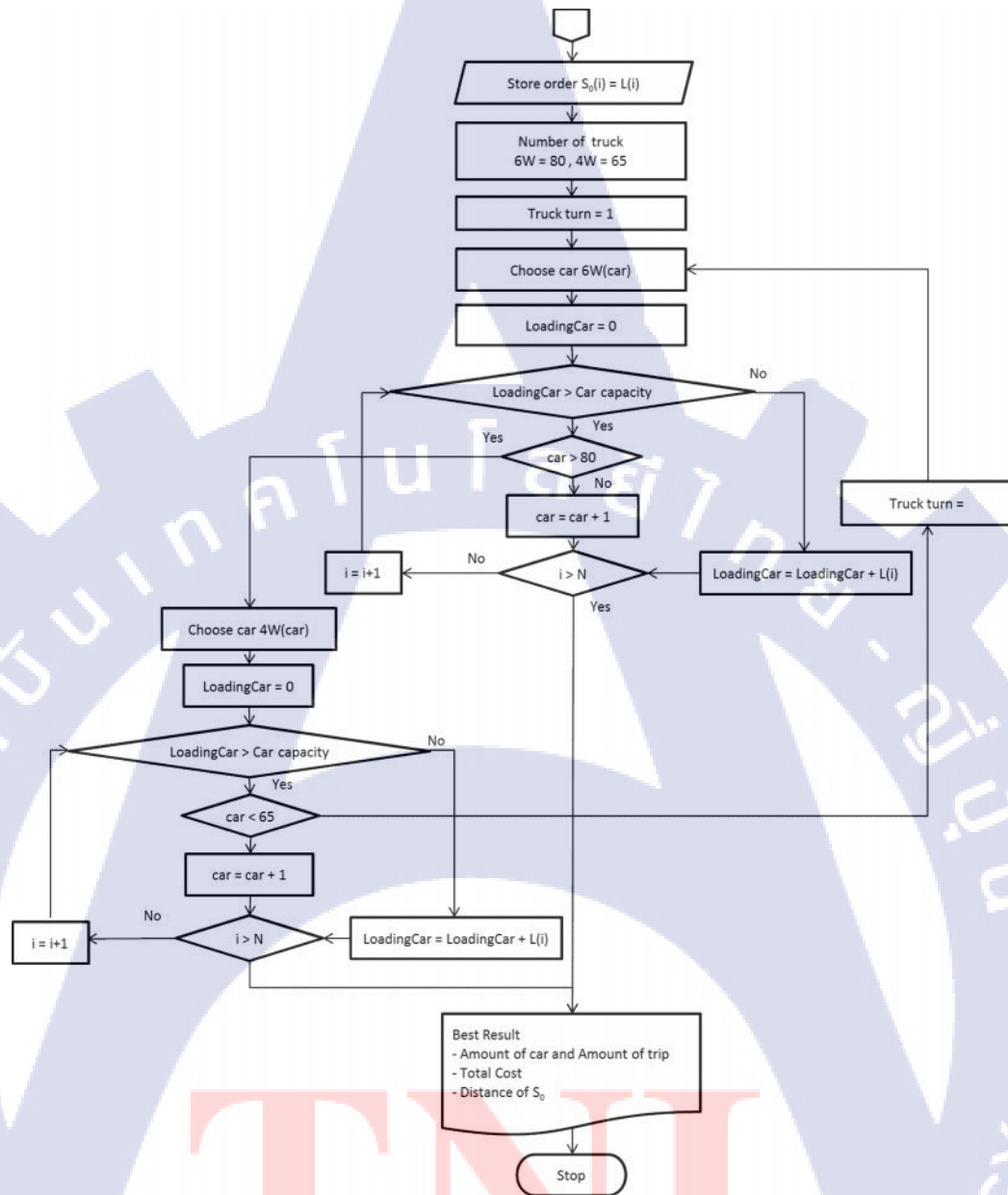


รูปที่ 3.11 หลักการการปรับปรุงเส้นทางแบบ SWAP-INSERT และคำนวณการใช้รถแบบ FTL (ต่อ)

3.4.4.9 การปรับปรุงเส้นทางแบบ INSERT-SWAP และคำนวณการใช้รถแบบ LTL



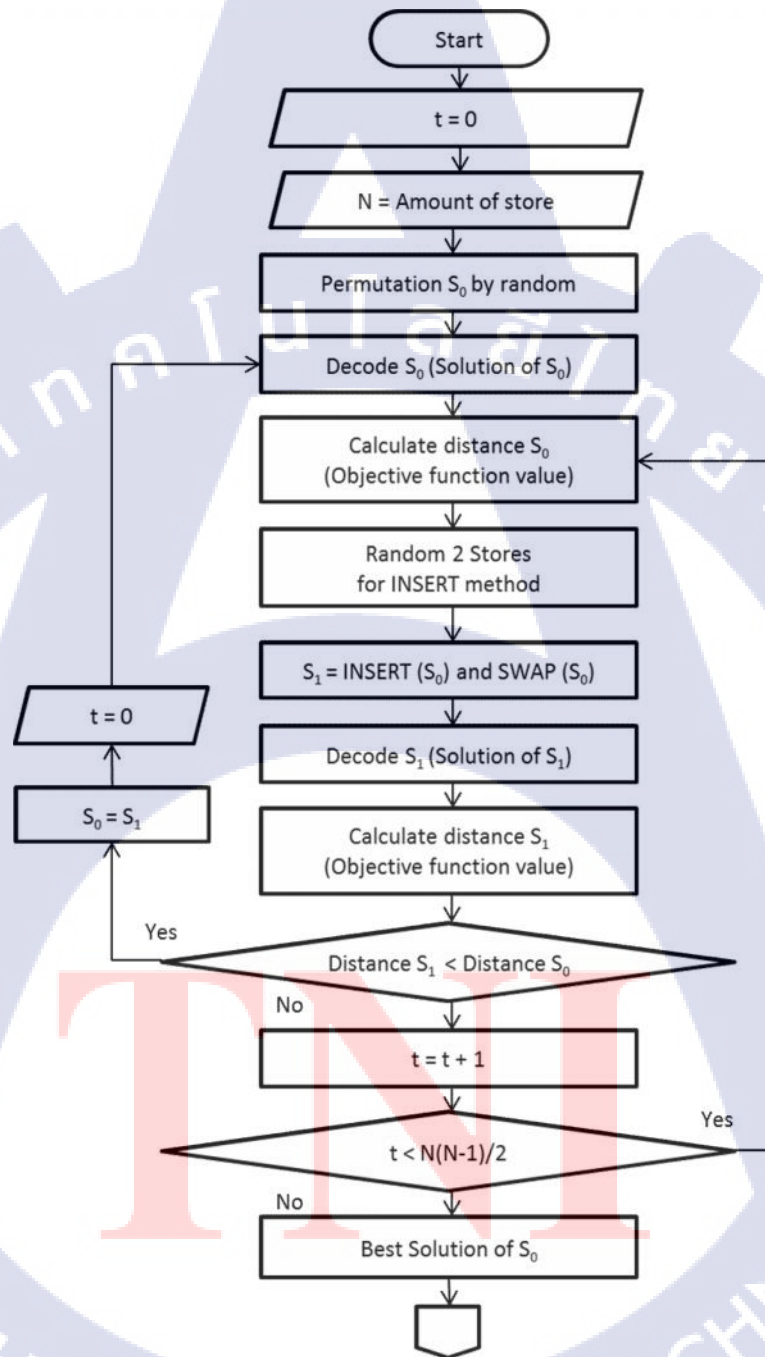
รูปที่ 3.12 หลักการการปรับปรุงเส้นทางแบบ INSERT-SWAP และคำนวณการใช้รถแบบ LTL



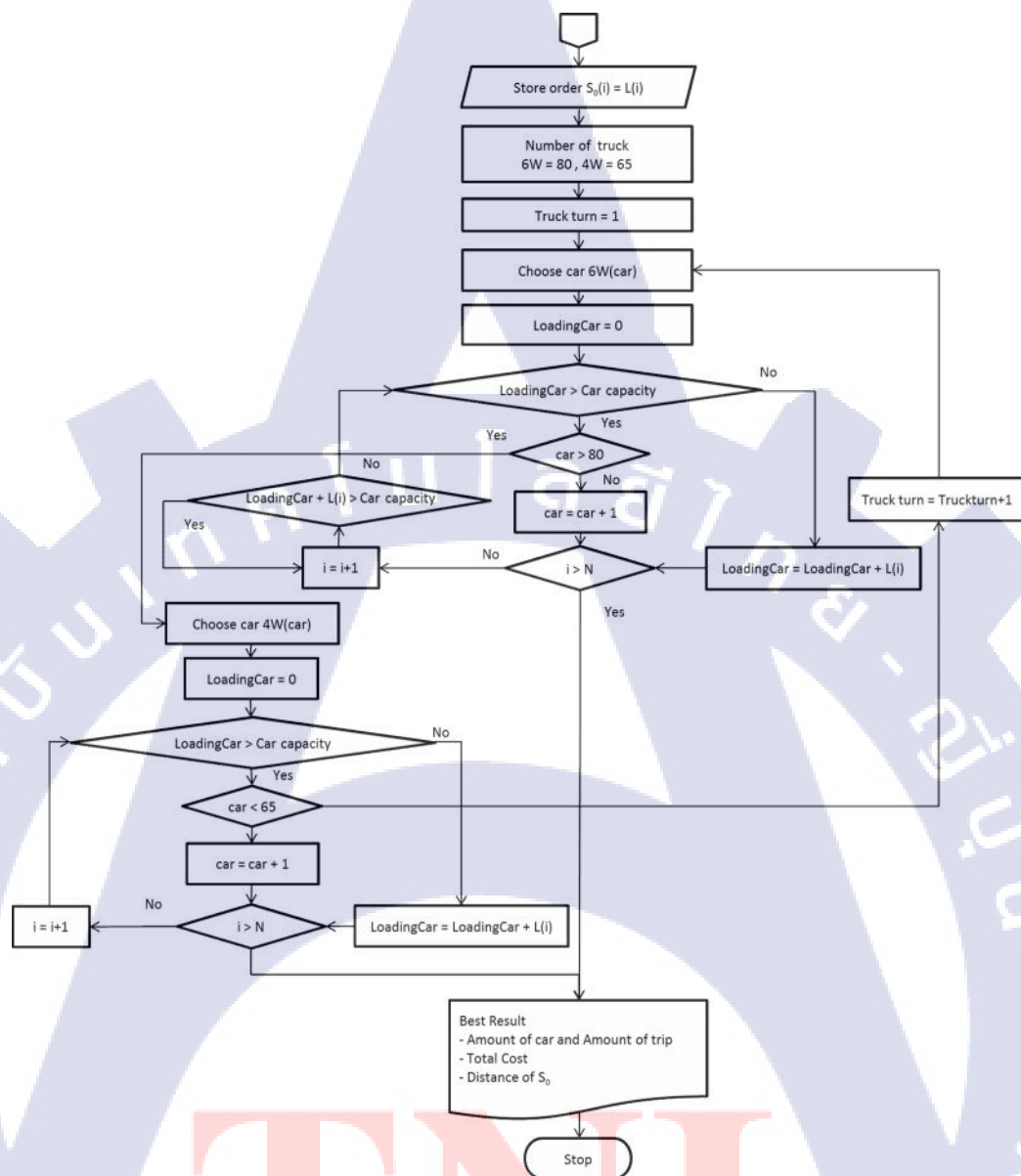
รูปที่ 3.12 หลักการการปรับปรุงเส้นทางแบบ INSERT-SWAP และคำนวณการใช้รถแบบ LTL (ต่อ)

3.4.4.10 การปรับปรุงเส้นทางแบบ INSERT-SWAP และคำนวณการใช้รถแบบ

FTL



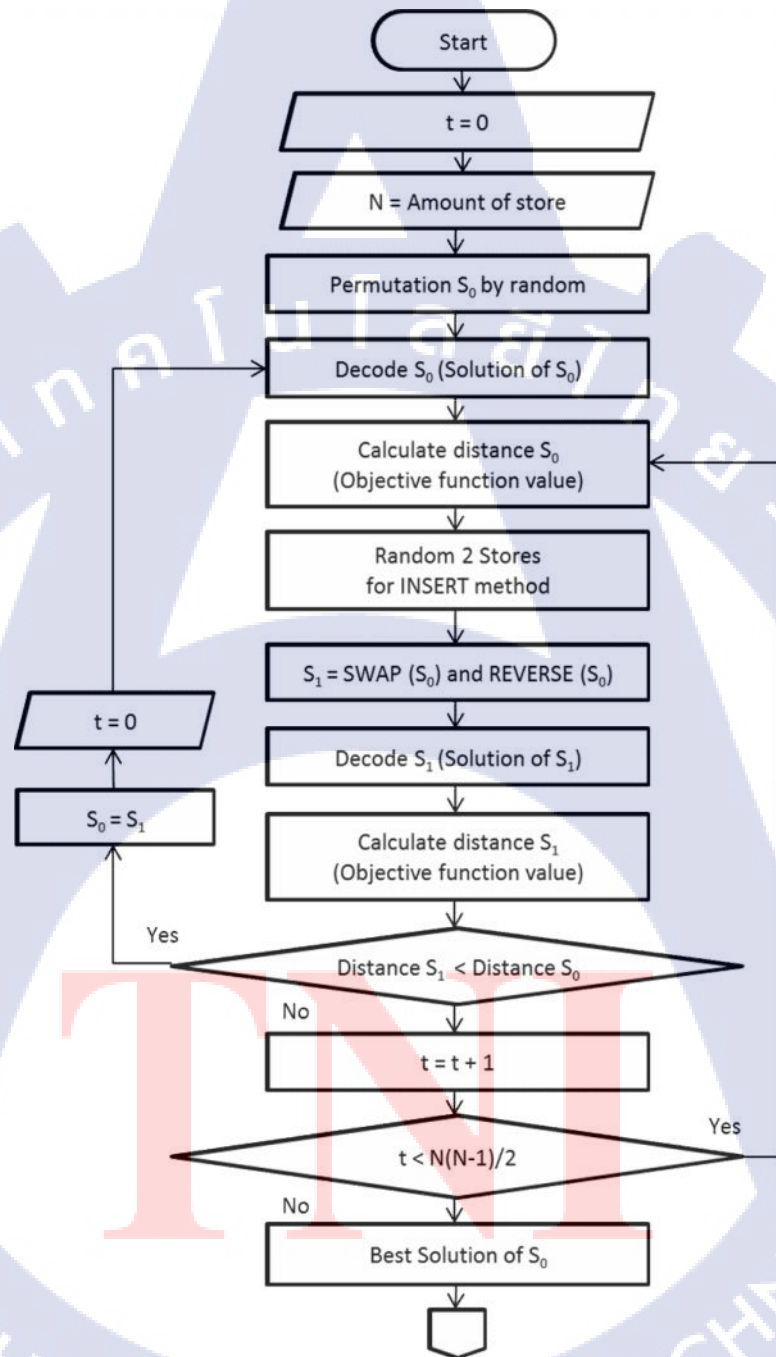
รูปที่ 3.13 หลักการการปรับปรุงเส้นทางแบบ INSERT-SWAP และคำนวณการใช้รถแบบ FTL



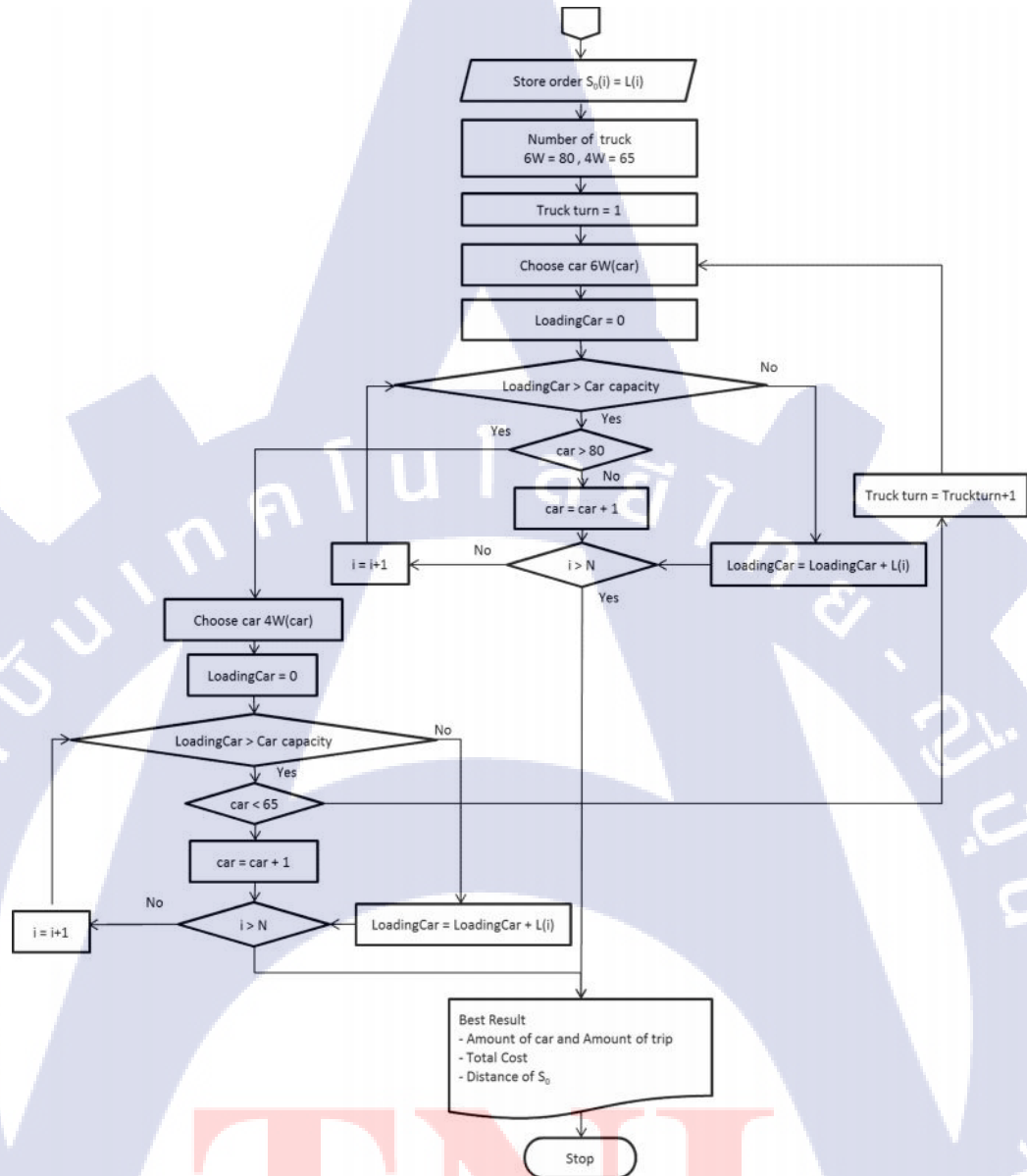
รูปที่ 3.13 หลักการการปรับปรุงเส้นทางแบบ INSERT-SWAP และคำนวณการใช้รถแบบ FTL (ต่อ)

3.4.4.11 การปรับปรุงเส้นทางแบบ SWAP-REVERSE และจำนวนการใช้รถแบบ

LTL



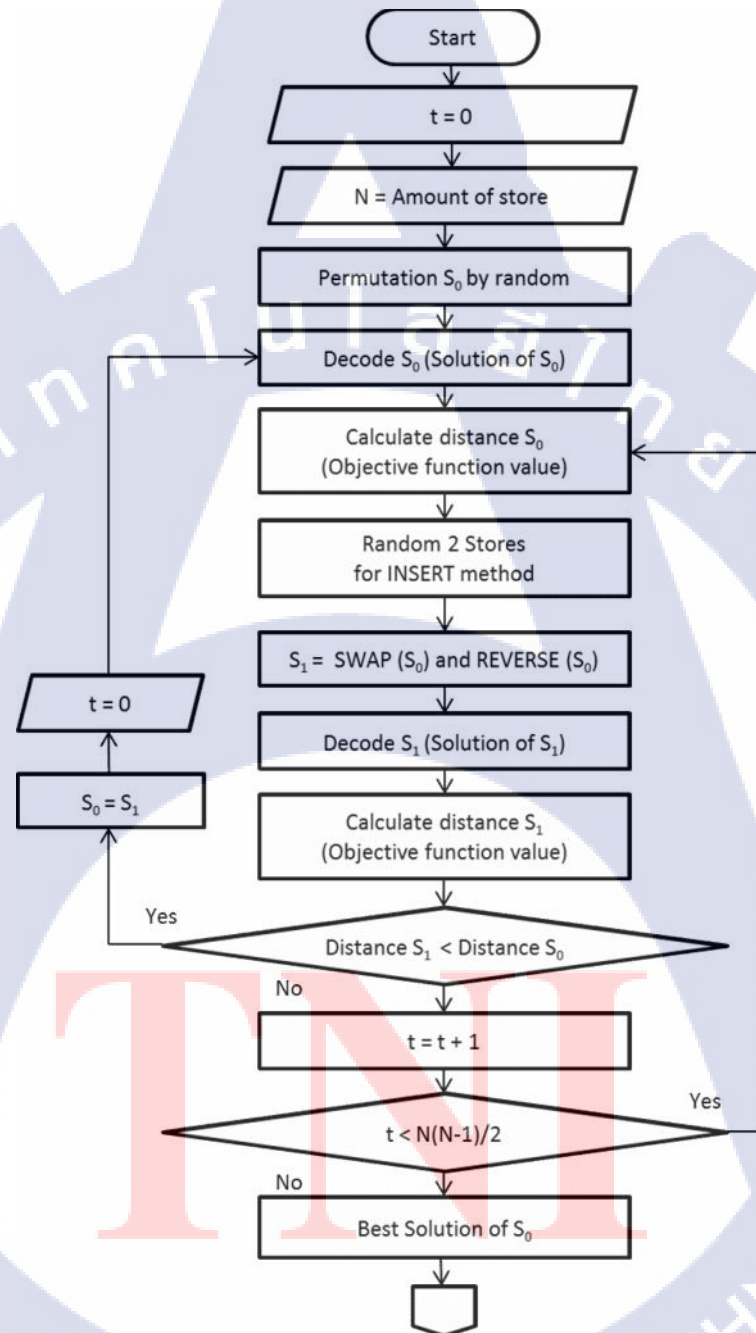
รูปที่ 3.14 หลักการการปรับปรุงเส้นทางแบบ SWAP-REVERSE และจำนวนการใช้รถแบบ LTL



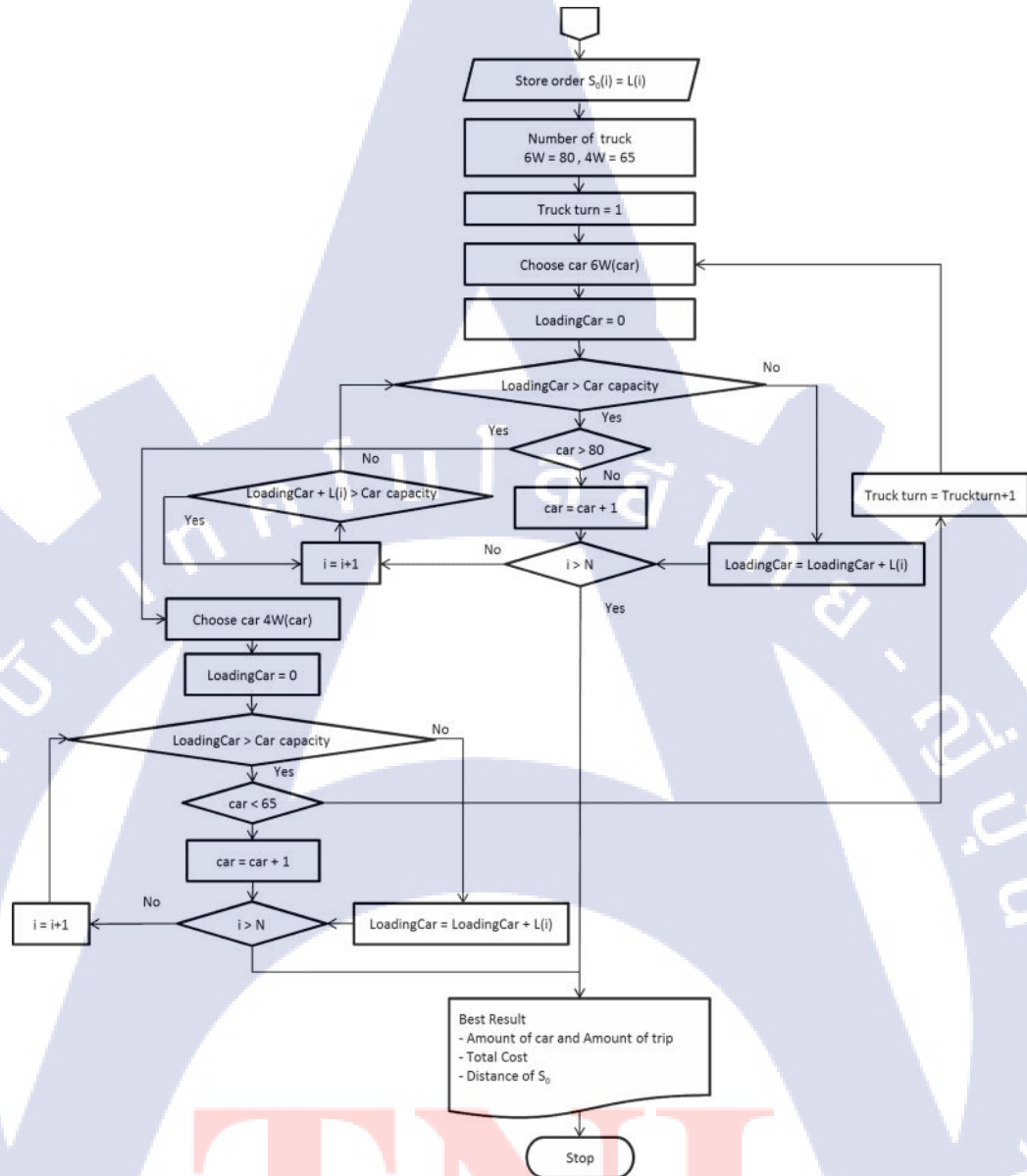
รูปที่ 3.14 หลักการการปรับปรุงเส้นทางแบบ SWAP-REVERSE และคำนวณการใช้รถแบบ LTL (ต่อ)

3.4.4.12 การปรับปรุงเส้นทางแบบ SWAP-REVERSE และคำนวณการใช้รถแบบ

FTL



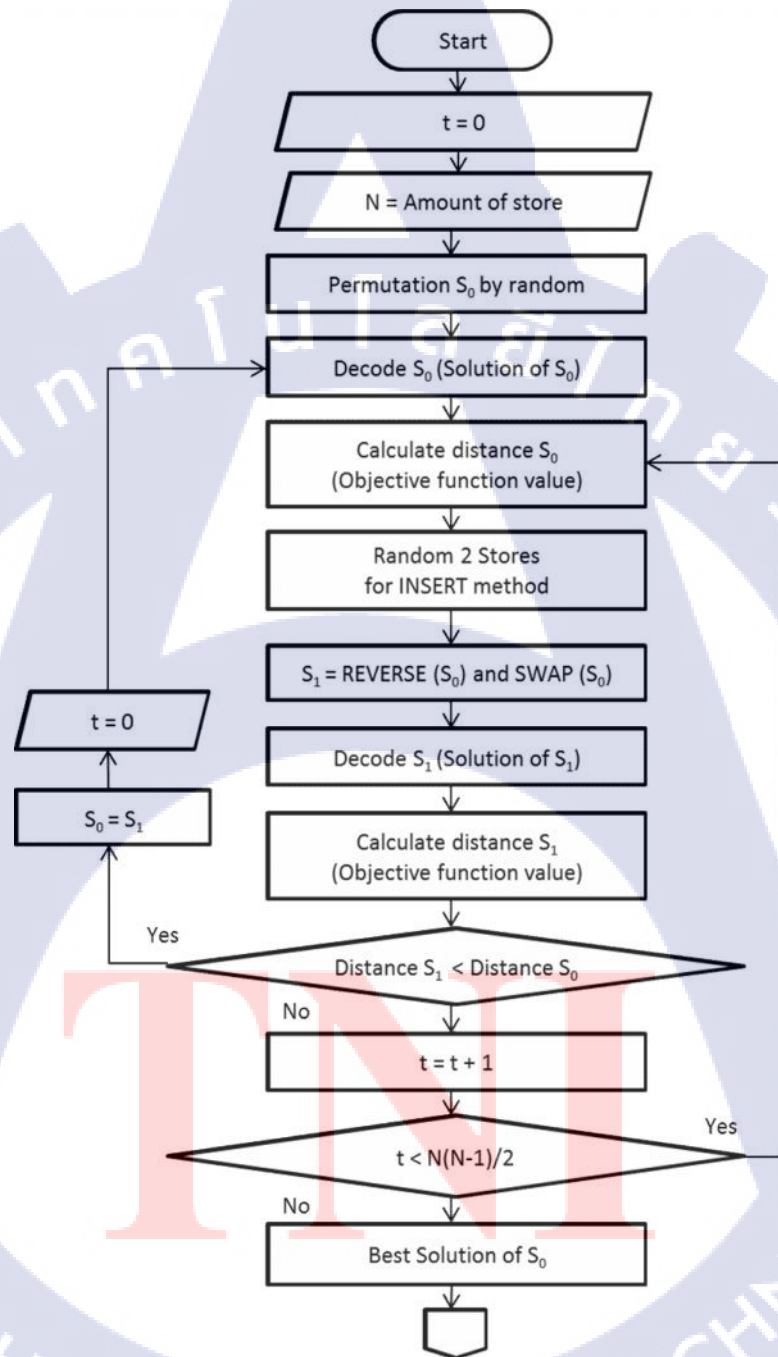
รูปที่ 3.15 หลักการการปรับปรุงเส้นทางแบบ SWAP-REVERSE และคำนวณการใช้รถแบบ FTL



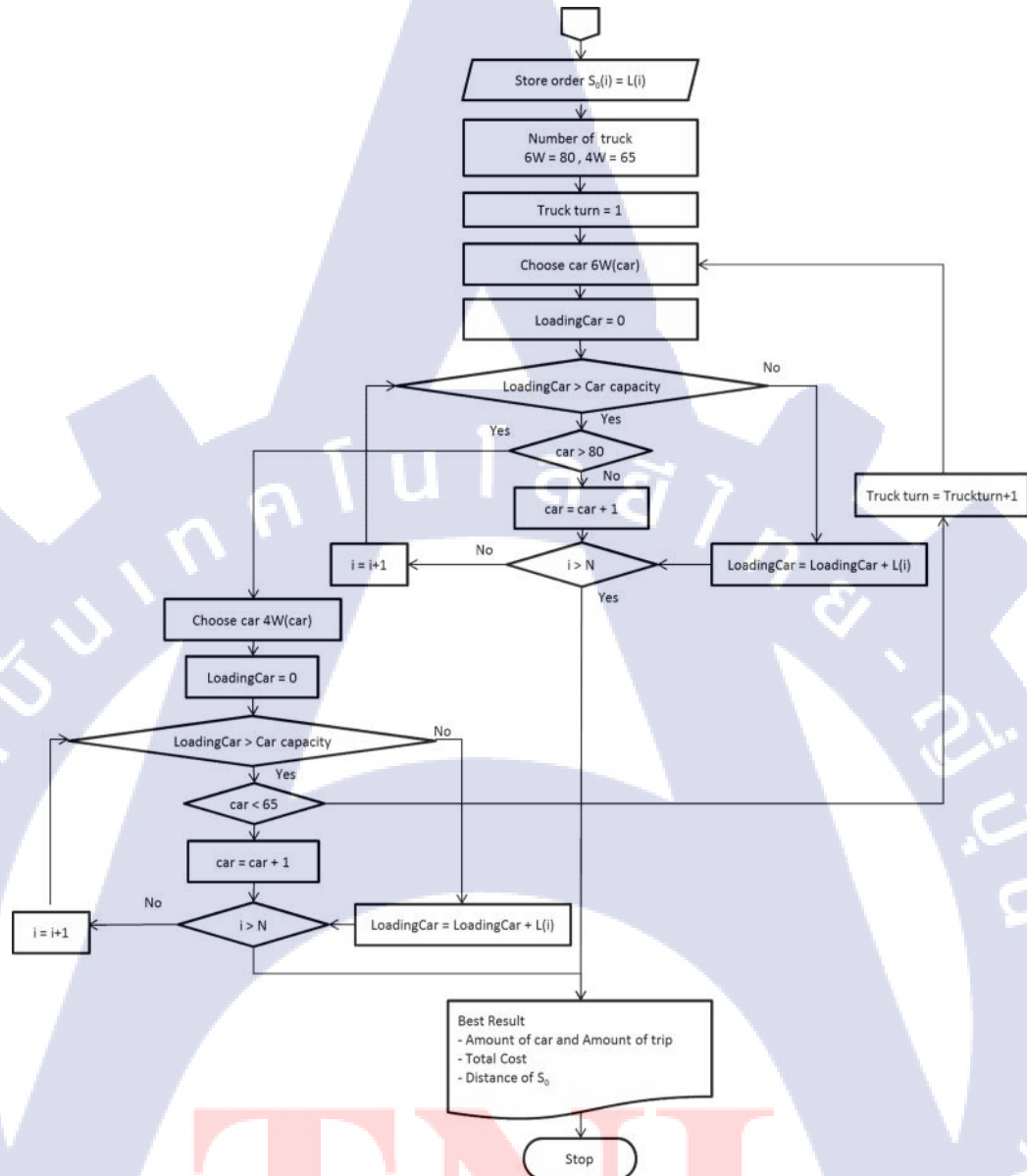
รูปที่ 3.15 หลักการการปรับปรุงเส้นทางแบบ SWAP-REVERSE และคำนวณการใช้รถแบบ FTL (ต่อ)

3.4.4.13 การปรับปรุงเส้นทางแบบ REVERSE-SWAP และจำนวนการใช้รถแบบ

LTL



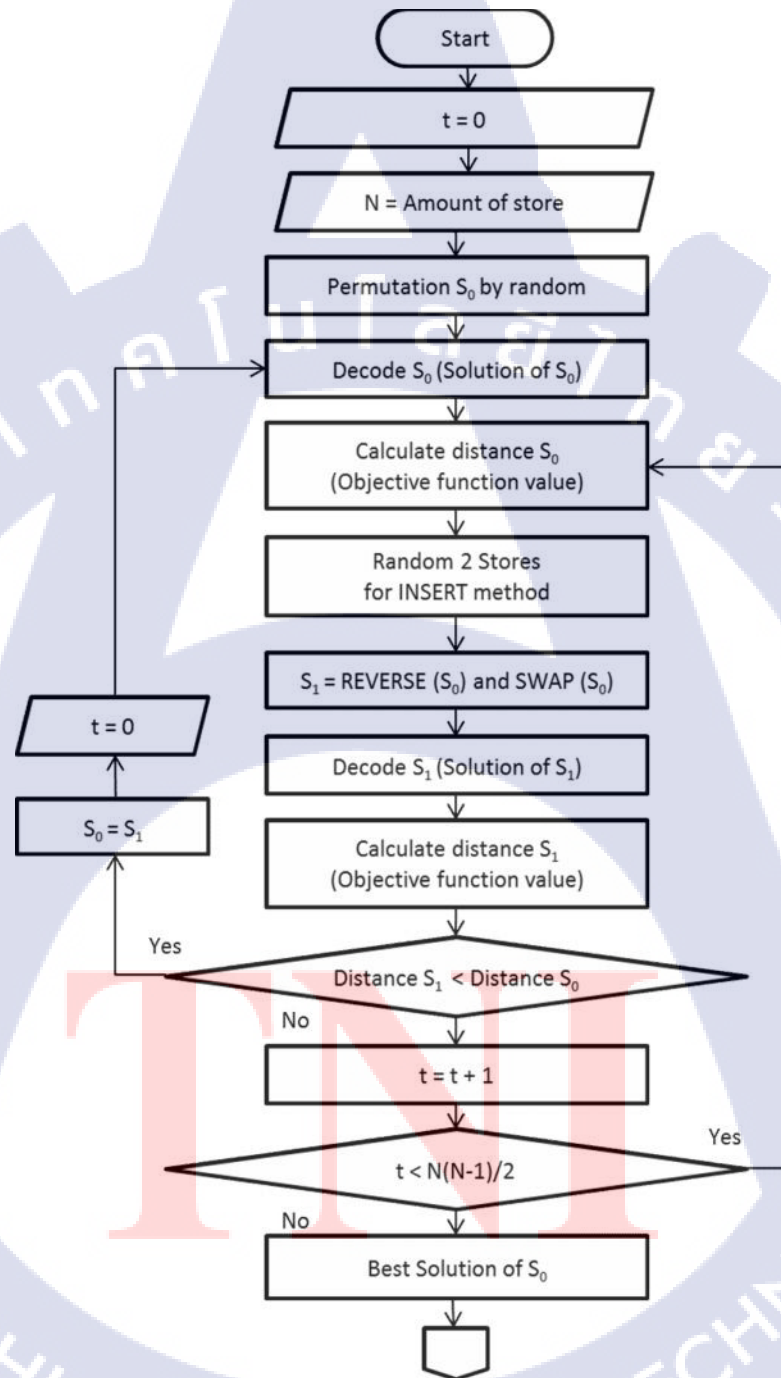
รูปที่ 3.16 หลักการการปรับปรุงเส้นทางแบบ REVERSE-SWAP และจำนวนการใช้รถแบบ LTL



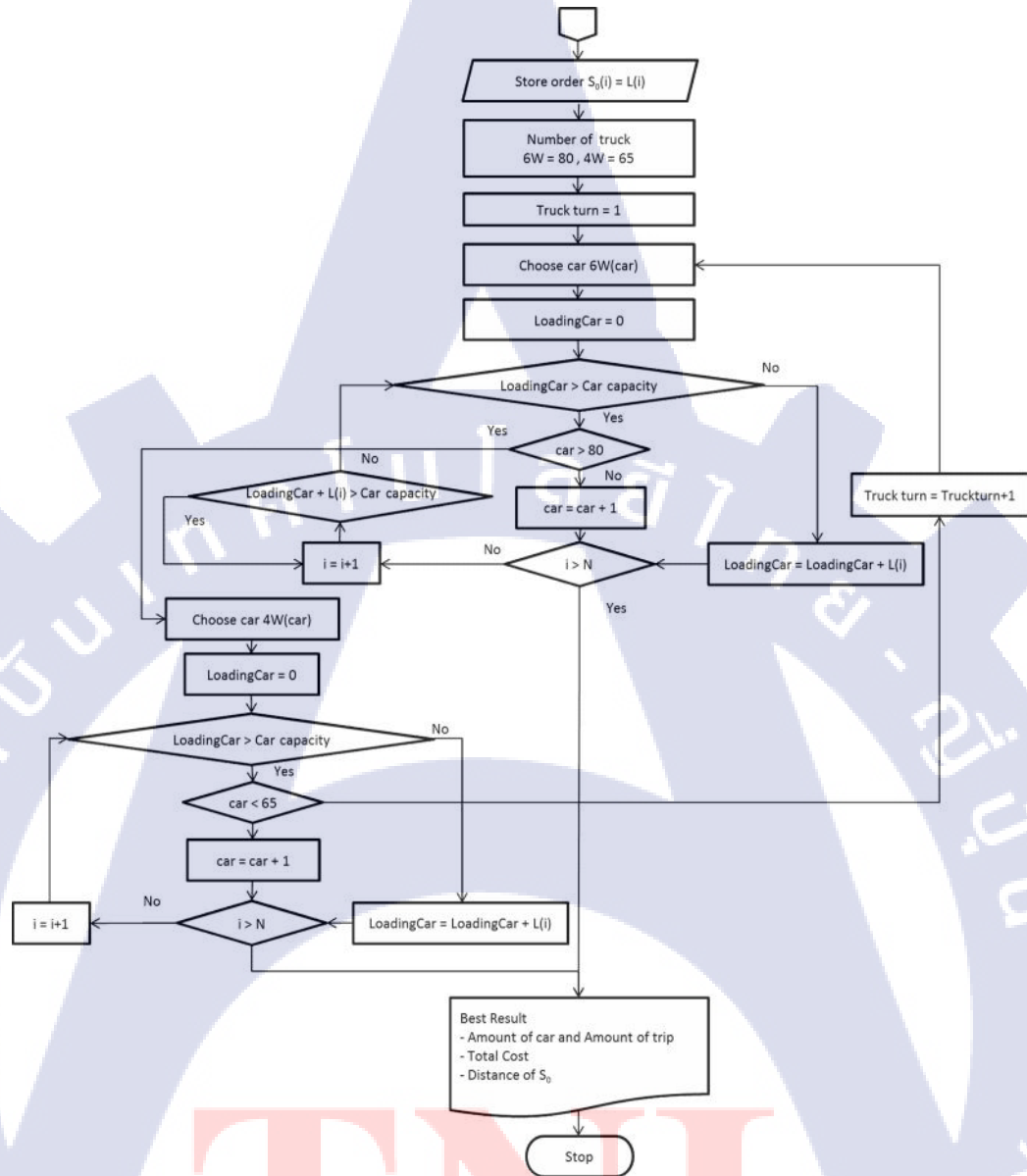
รูปที่ 3.16 หลักการการปรับปรุงเส้นทางแบบ REVERSE-SWAP และคำนวณการใช้รถแบบ LTL (ต่อ)

3.4.4.14 การปรับปรุงเส้นทางแบบ REVERSE-SWAP และจำนวนการใช้รถแบบ

FTL



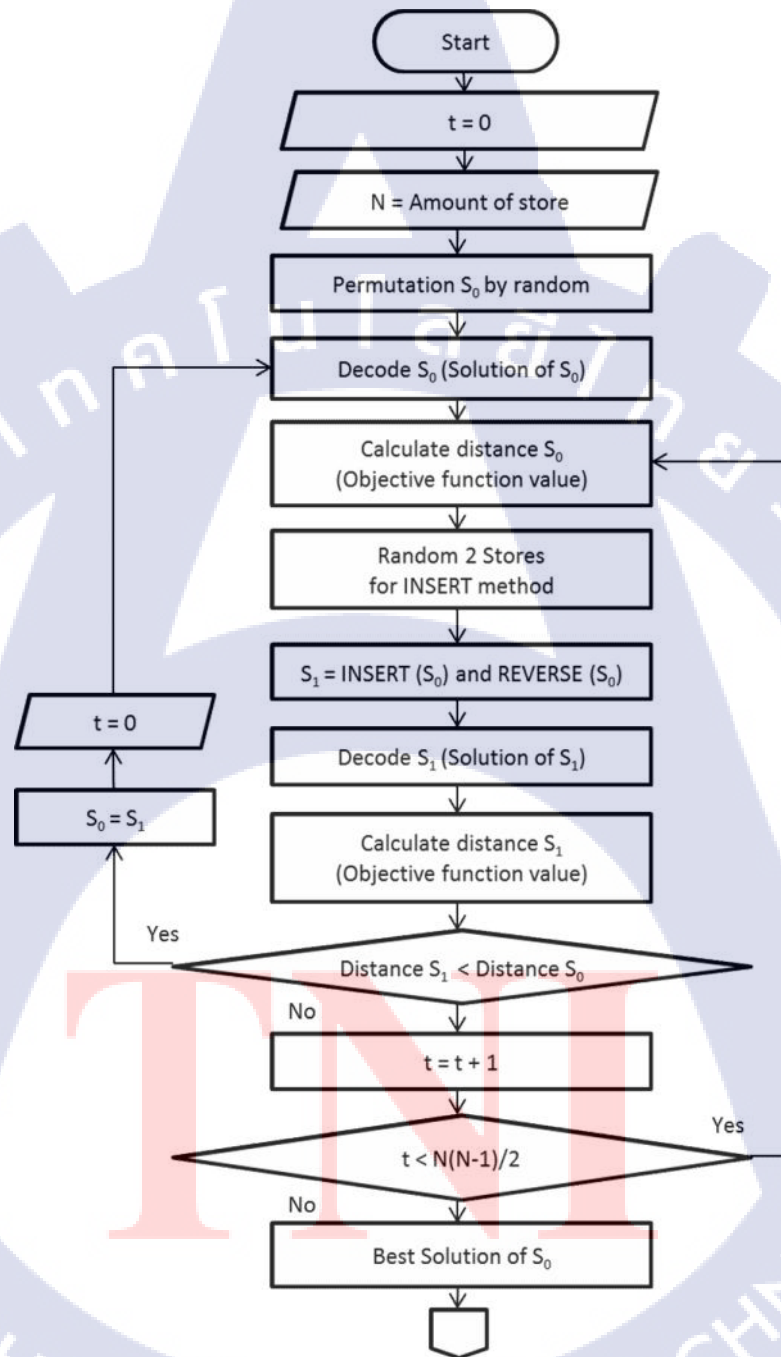
รูปที่ 3.17 หลักการการปรับปรุงเส้นทางแบบ REVERSE-SWAP และจำนวนการใช้รถแบบ FTL



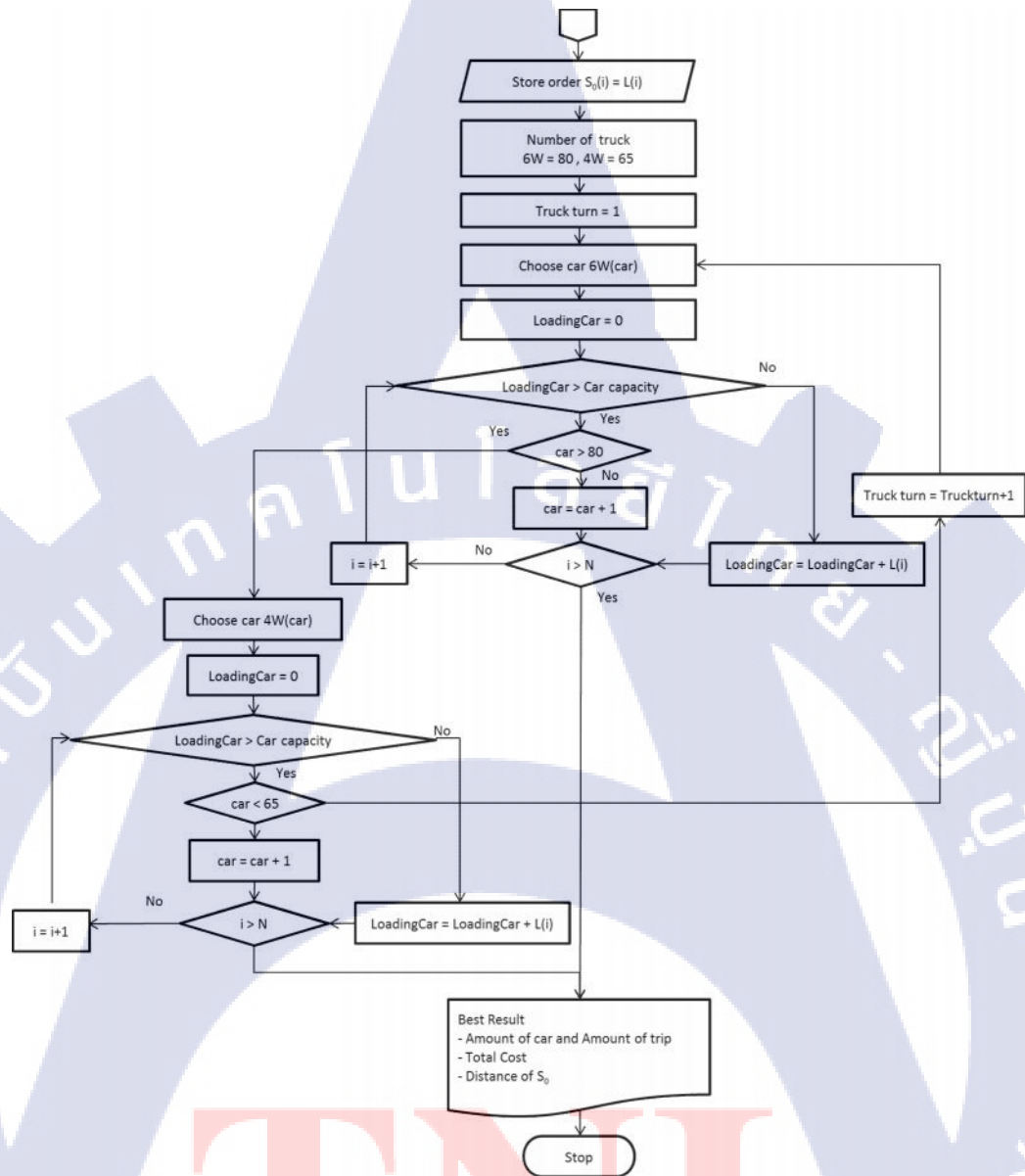
รูปที่ 3.17 หลักการการปรับปรุงเส้นทางแบบ REVERSE-SWAP และคำนวณการใช้รถแบบ FTL (ต่อ)

3.4.4.15 การปรับปรุงเส้นทางแบบ INSERT-REVERSE และคำนวณการใช้รถแบบ

LTL



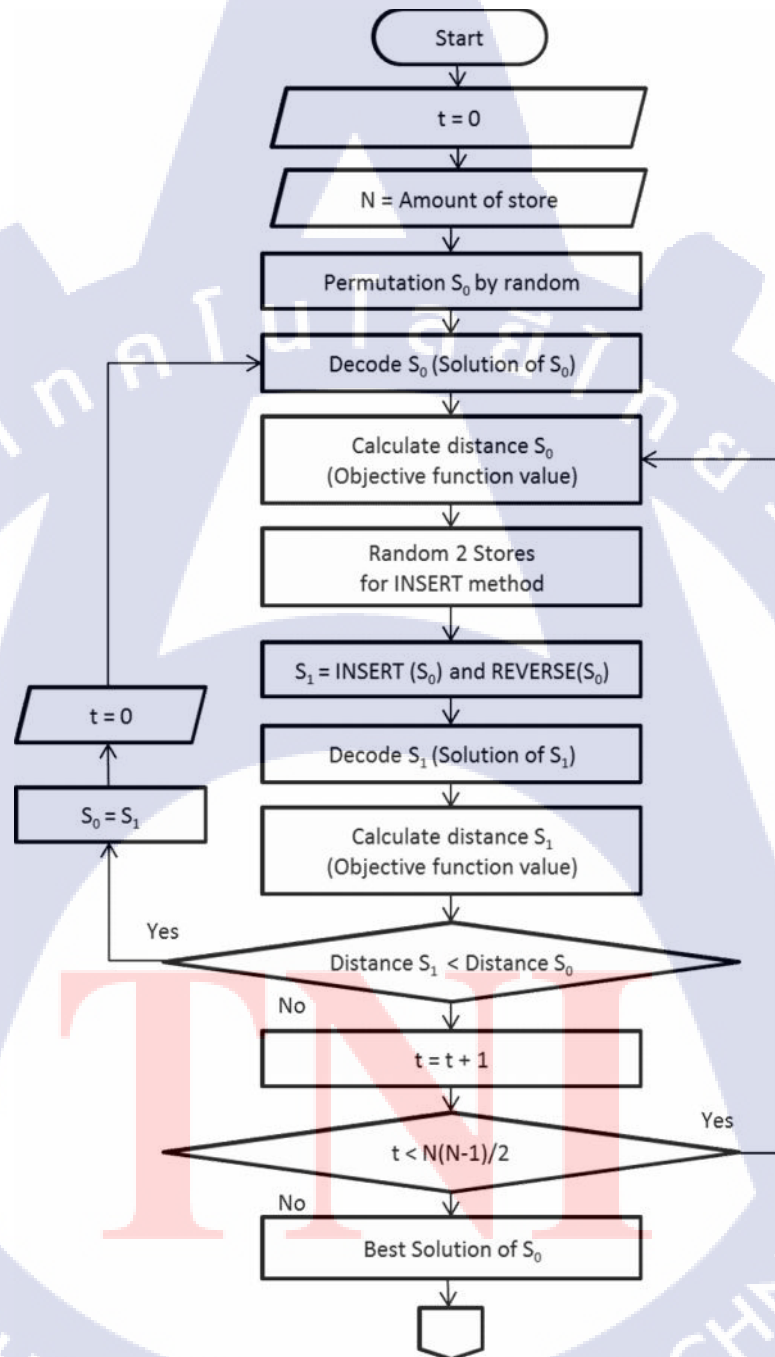
รูปที่ 3.18 หลักการการปรับปรุงเส้นทางแบบ INSERT-REVERSE และคำนวณการใช้รถแบบ LTL



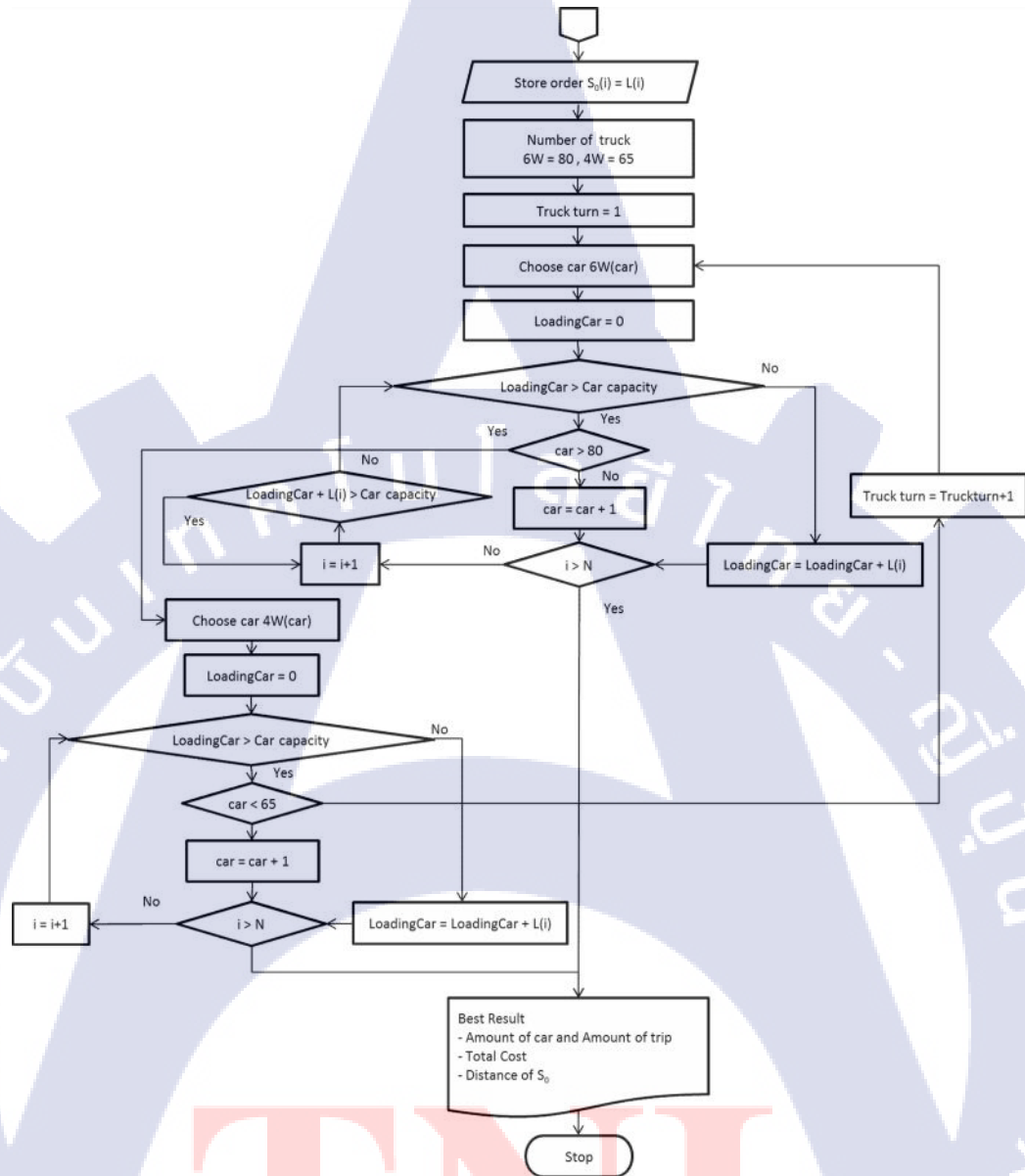
รูปที่ 3.18 หลักการการปรับปรุงเส้นทางแบบ INSERT-REVERSE และคำนวณการใช้รถแบบ LTL (ต่อ)

3.4.4.16 การปรับปรุงเส้นทางแบบ INSERT-REVERSE และคำนวณการใช้รถแบบ

FTL



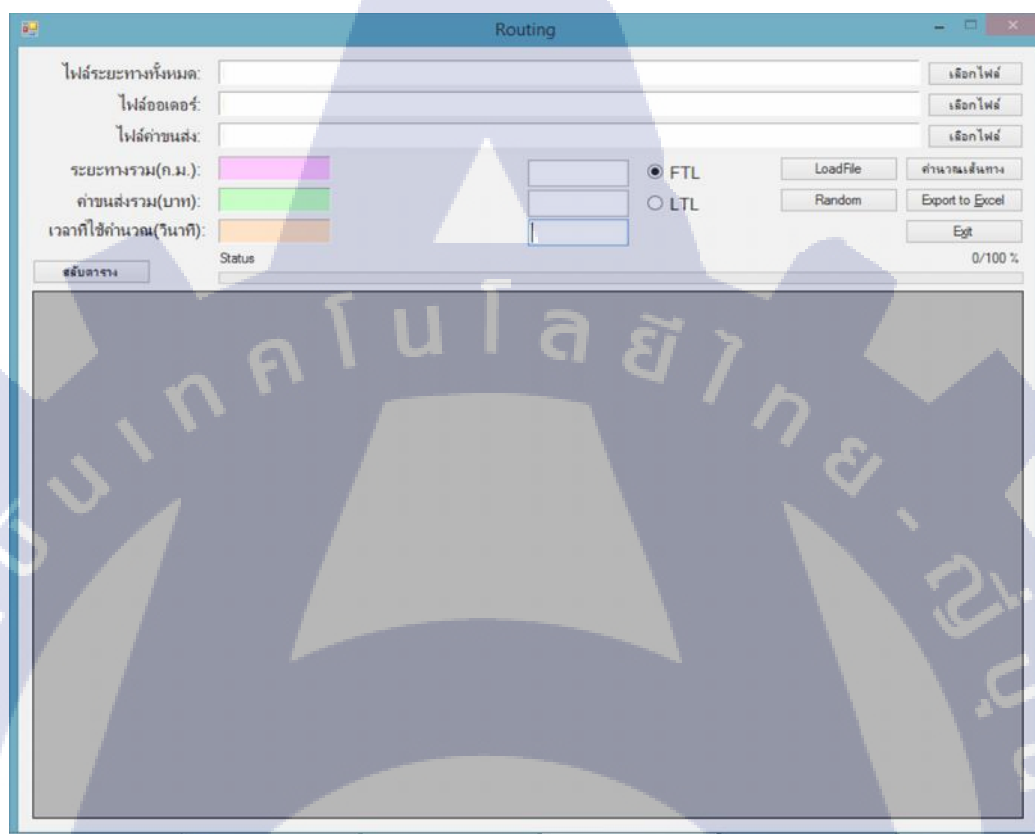
รูปที่ 3.19 หลักการการปรับปรุงเส้นทางแบบ INSERT-REVERSE และคำนวณการใช้รถแบบ FTL



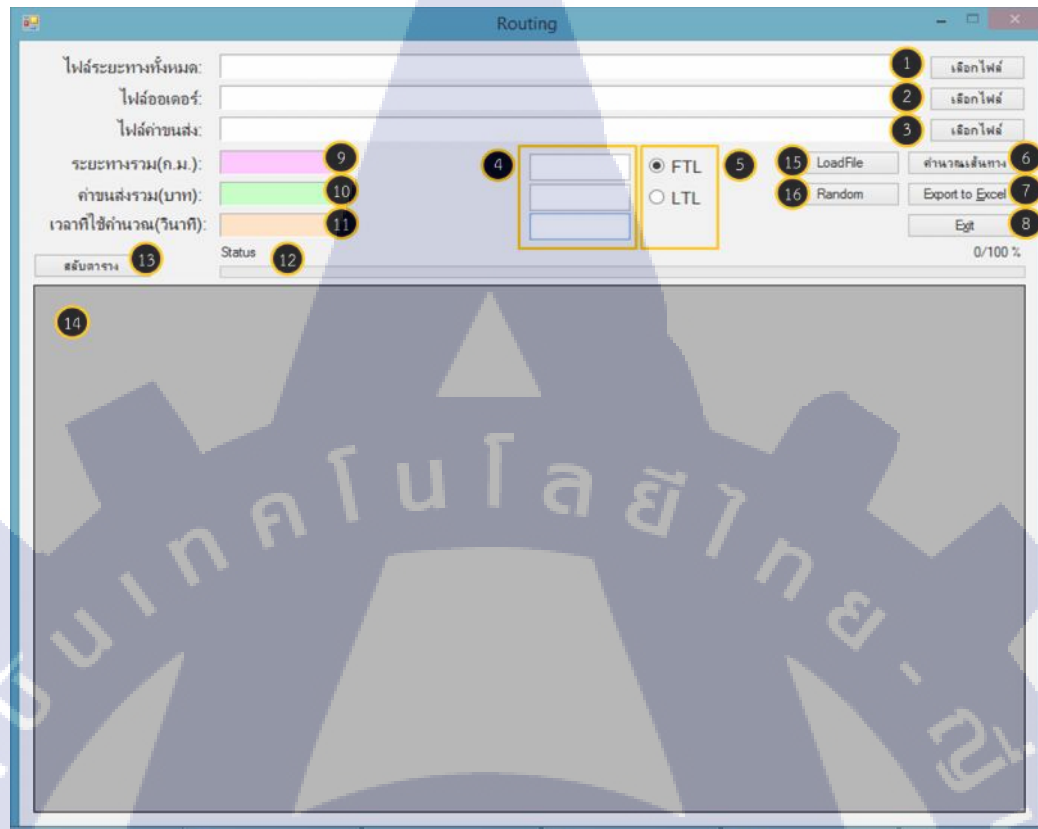
รูปที่ 3.19 หลักการการปรับปรุงเส้นทางแบบ INSERT-REVERSE และคำนวณการใช้รถแบบ FTL (ต่อ)

3.4.5 การทำงานของโปรแกรม

3.4.5.1 หน้าจอหลักโปรแกรม



รูปที่ 3.20 หน้าจอหลัก



รูปที่ 3.21 อธิบายส่วนประกอบของหน้าจอหลัก

จุดที่ 1 สำหรับเลือกไฟล์ระยะทาง (.csv)

จุดที่ 2 สำหรับเลือกไฟล์ Order หรือความต้องการสินค้าของร้าน (.xls)

จุดที่ 3 สำหรับเลือกไฟล์ ค่าขนส่ง (.xls)

จุดที่ 4 สำหรับเลือกรูปแบบการคำนวณ

- Swap คือ สุ่มเลือก 2 ร้าน และทำการสลับตำแหน่ง เพื่อคำนวณหาเส้นทางใหม่ที่มีค่าขนส่งต่ำกว่า
- Insert คือ สุ่มเลือก 1 ร้าน และตำแหน่งสำหรับการแทรก เพื่อคำนวณหาเส้นทางใหม่ที่มีค่าขนส่งต่ำกว่า
- Reverse คือ สุ่มเลือก 2 ร้าน และกลับตำแหน่ง เพื่อคำนวณหาเส้นทางใหม่ที่มีค่าขนส่งต่ำกว่า

จุดที่ 5 สำหรับเลือกรูปแบบการคำนวณการบรรทุก

- FTL คือ Full Truck Load (บรรทุกของเต็มรถ)
- LTL คือ Less than Truck Load (บรรทุกไม่เต็มรถ)

จุดที่ 6 สำหรับเริ่มคำนวณเส้นทางตามวิธีที่เลือก

จุดที่ 7 สำหรับ Export ข้อมูลการคำนวณที่ได้ ออกเป็นรูปแบบของ Excel File

จุดที่ 8 สำหรับออกจากโปรแกรม

จุดที่ 9 แสดงผลระยะทางรวมที่ได้จากการคำนวณ หน่วย กิโลเมตร

จุดที่ 10 แสดงผลค่าขนส่งรวมที่ได้จากการคำนวณ หน่วย บาท

จุดที่ 11 แสดงเวลาที่ใช้ประมวลผล หน่วย วินาที

จุดที่ 12 แถบเส้นแสดงสถานะการประมวลผล

จุดที่ 13 สำหรับปรับรูปแบบตารางแสดงผล

จุดที่ 14 ตารางแสดงผล

จุดที่ 15 สำหรับกด Load ไฟล์ข้อมูลที่เลือกไว้ (กดหลังจากเลือกไฟล์แต่ละครั้ง)

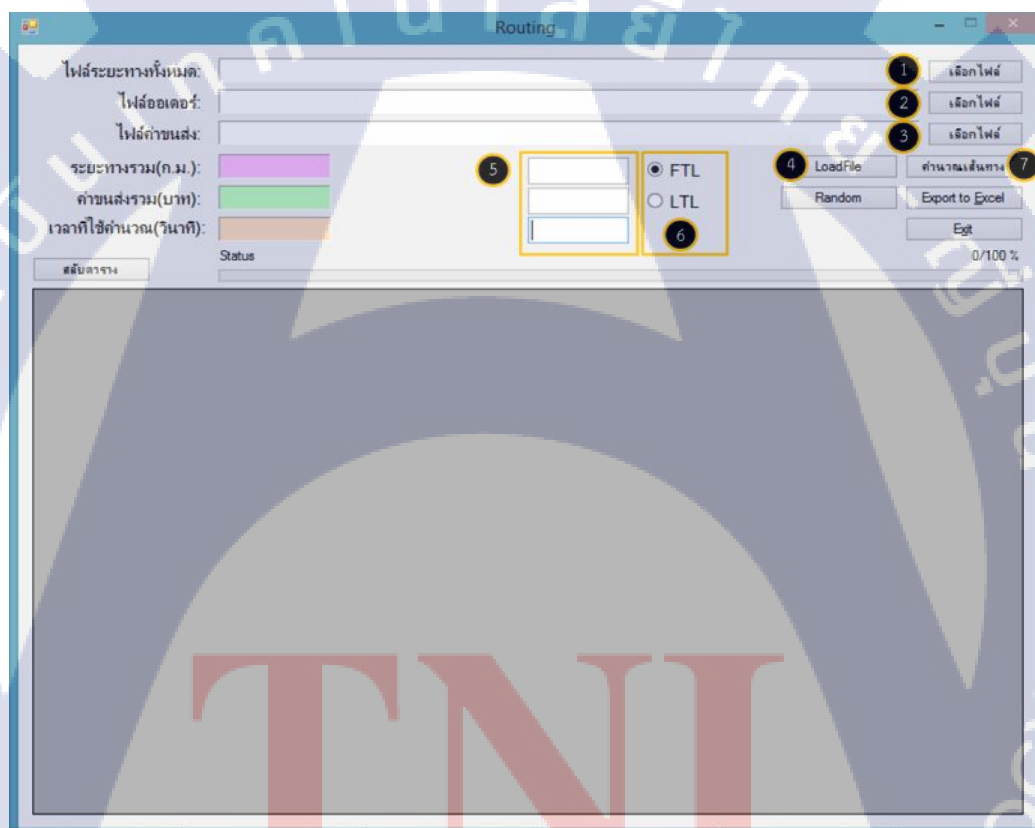
จุดที่ 16 สำหรับกรณีต้องการจัดสายแบบสุ่ม ใหม่

TNI

THAI - NICHI INSTITUTE OF TECHNOLOGY

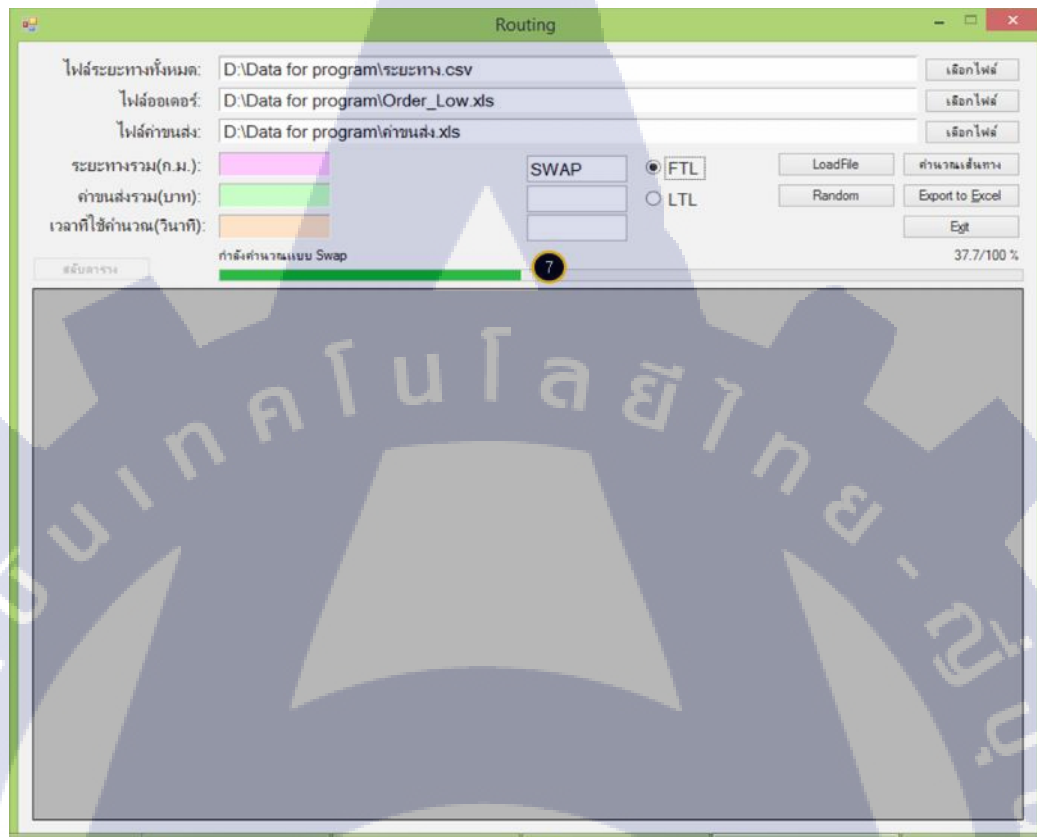
3.4.5.2 การใช้งานโปรแกรม

1. เลือกไฟล์ระยะทาง (.csv)
2. เลือกไฟล์ Order หรือความต้องการสินค้าของร้าน (.xls)
3. เลือกไฟล์ ค่าขนส่ง (.xls)
4. กดปุ่ม "Load File"
5. ใส่รูปแบบการคำนวณ (SWAP/INSERT/REVERSE)
6. เลือกรูปแบบการคำนวณการบรรจุทุก (เลือกได้เพียง 1)
7. กดปุ่ม "คำนวณเส้นทาง" เพื่อคำนวณเส้นทางตามวิธีที่เลือก



รูปที่ 3.22 อธิบายการใช้งานโปรแกรม รูปที่ 1

7. รอกการประมวลผล



รูปที่ 3.23 อธิบายการใช้งานโปรแกรม รูปที่ 2

8. การประมวลผลเสร็จสิ้น

Routing

ไฟล์ระยะทางทั้งหมด: D:\Data for program\ระยะทาง.csv
 ไฟล์ออเดอร์: D:\Data for program\Order.xls
 ไฟล์คำนวณส่ง: D:\Data for program\คำนวณส่ง.xls

ระยะทางรวม(ก.ม.): 204102.92
 ค่าขนส่งรวม(บาท): 357906.25
 เวลาที่ใช้คำนวณ(วินาที): 3388.63

SWAP ☒ FTL ☐ LTL

LoadFile Random Export to Excel Exit

คำนวณเสร็จเรียบร้อยแล้ว 100.0/100 %

No.	Truck No.	Truck Type	Truck Size	Shop Name	Sub Volume	Total Volume	Distance From DC	Transport Fee1	Transport Fee2
1	1	4W	5.50				217.36	547.50	615.00
2							264.45	615.00	
3							186.75	475.00	
4	2	4W	5.50				273.59	615.00	615.00
5							119.49	463.50	
6							132.76	417.50	
7	3	4W	5.50				217.09	565.00	565.00
8	4	4W	5.50				184.68	503.75	503.75
9	5	4W	5.50	AAR	4.00	5.00	243.71	602.50	615.00
10				ADF	1.00		269.58	615.00	
11	6	4W	5.50	LQ	2.00	5.00	185.55	475.00	475.00
12				VY	3.00		20.13	190.50	
13	7	4W	5.50	GL	5.00	5.00	135.93	475.00	475.00
14	8	4W	5.50	TH	5.00	5.00	313.21	670.75	670.75
15	9	4W	5.50	XR	3.00	5.00	286.23	677.50	677.50
16				J	2.00		269.89	615.00	
17	10	4W	5.50	QU	3.00	5.00	265.93	615.00	615.00
18				BY	1.00		55.93	268.75	
19				TK	1.00		208.53	567.50	

รูปที่ 3.24 อธิบายการใช้งานโปรแกรม รูปที่ 3

9. แสดงผลการคำนวณในตาราง

10. คลิก "Export to Excel" เพื่อ Export ข้อมูลออกมาในรูปแบบ Excel

Routing

ไฟล์ระยะทางทั้งหมด: D:\Data for program\ระยะทาง.csv
 ไฟล์ออเดอร์: D:\Data for program\Order.xls
 ไฟล์คำนวณ: D:\Data for program\คำนวณ.xls

ระยะทางรวม(ก.ม.): 204102.92
 ค่าขนส่งรวม(บาท): 357906.25
 เวลาที่ใช้คำนวณ(วินาที): 3388.63

SWAP ☒ FTL ☐ LTL

LoadFile Random Export to Excel

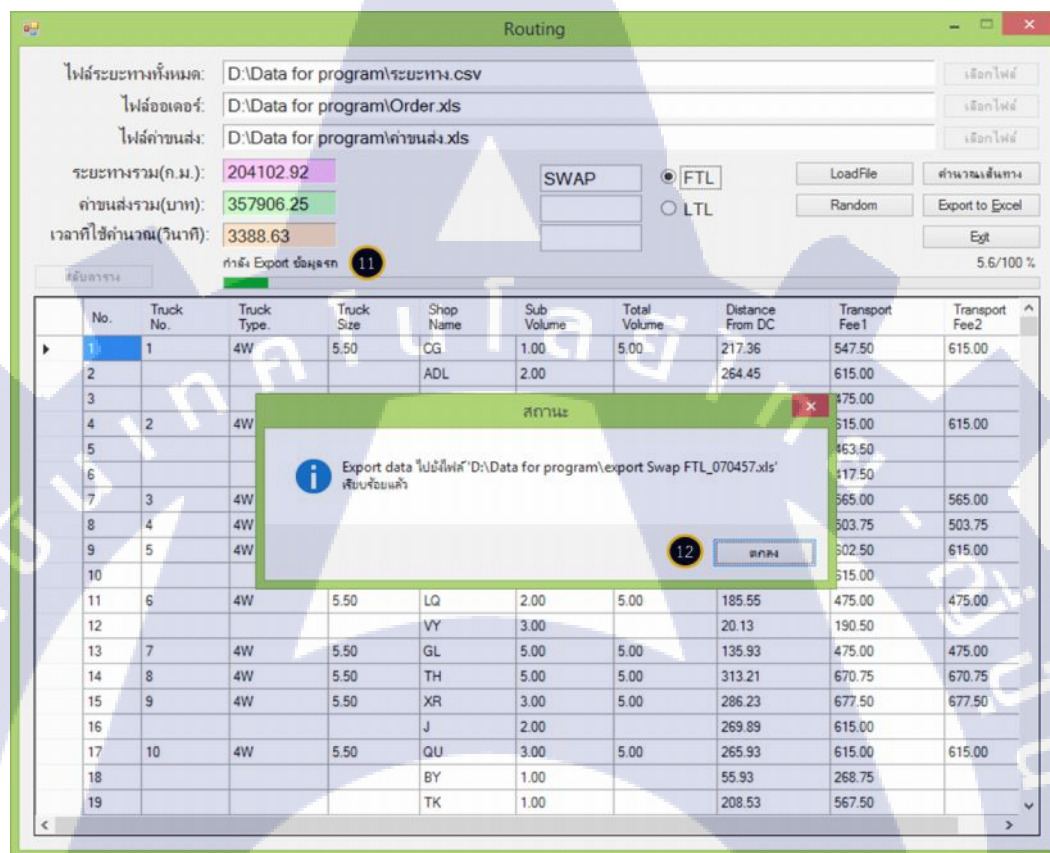
คำนวณเสร็จเรียบร้อยแล้ว 100.0/100 %

No.	Truck No.	Truck Type	Truck Size	Shop Name	Sub Volume	Total Volume	Distance From DC	Transport Fee1	Transport Fee2
1	1	4W	5.50	CG	1.00	5.00	217.36	547.50	615.00
2				ADL	2.00		264.45	615.00	
3				GN	2.00		186.75	475.00	
4	2	4W	5.50	ZO	2.00	5.00	273.59	615.00	615.00
5				ZH	2.00		119.49	463.50	
6				CZ	1.00		132.76	417.50	
7	3	4W	5.50	ADT	5.00	5.00	217.09	565.00	565.00
8	4	4W	5.50	WN	5.00	5.00	184.68	503.75	503.75
9	5	4W	5.50	AAR	4.00	5.00	243.71	602.50	615.00
10				ADF	1.00		269.58	615.00	
11	6	4W	5.50	LQ	2.00	5.00	185.55	475.00	475.00
12				VY	3.00		20.13	190.50	
13	7	4W	5.50	GL	5.00	5.00	135.93	475.00	475.00
14	8	4W	5.50	TH	5.00	5.00	313.21	670.75	670.75
15	9	4W	5.50	XR	3.00	5.00	286.23	677.50	677.50
16				J	2.00		269.89	615.00	
17	10	4W	5.50	QU	3.00	5.00	265.93	615.00	615.00
18				BY	1.00		55.93	268.75	
19				TK	1.00		208.53	567.50	

รูปที่ 3.25 อธิบายการใช้งานโปรแกรม รูปที่ 4

11. แสดงผลการ Export ข้อมูลในตารางเป็น Excel

12. เมื่อ Export ข้อมูลเสร็จ ให้คลิก ตกลง



รูปที่ 3.26 อธิบายการใช้งานโปรแกรม รูปที่ 5

3.4.5.3 การแสดงผล

รูปแบบข้อมูลที่แสดงผลในรูปแบบของไฟล์ excel ซึ่งมีการแสดงผลออกเป็น

2 ซีท

ซีทที่ 1 (ชื่อซีท ค่าขนส่ง) ประกอบด้วย ข้อมูลดังนี้

- ระยะทางรวม (ก.ม.) แสดงผลการคำนวณระยะทางรวมในหน่วยกิโลเมตร
- ค่าขนส่งรวม (บาท) แสดงผลการคำนวณค่าขนส่งรวมในหน่วยบาท
- เวลาที่ใช้คำนวณ (วินาที) แสดงผลเวลาที่ใช้ในการคำนวณในหน่วยวินาที
- การคำนวณระยะทาง แสดงรูปแบบการคำนวณที่เลือก

- การคำนวณรถ แสดงรูปแบบการบรรทุกสินค้า
- รายการลำดับการขนส่งทั้งหมด

ลำดับที่	ลำดับการขนส่ง
รอบที่	รอบการวิ่งของรถ
หมายเลขรถ	หมายเลขของรถ
ชนิดของรถ	ประเภทรถ (4 ล้อ หรือ 6 ล้อ)
ชื่อร้าน	ชื่อร้านตามลำดับการขนส่ง
ความจุที่ออเดอร์	ความต้องการสินค้าของร้าน
ความจุรวม	ความจุรวมของรถ
ระยะทาง	ระยะทางที่เกิดขึ้นระหว่างต้นทางและปลายทาง
ค่าขนส่ง	ค่าขนส่งตามพื้นที่ของตำแหน่งร้าน
ค่าขนส่งที่คิด	ค่าขนส่งที่นำมาคิดของเส้นทางรถ โดยเลือกจาก ค่าขนส่งตามพื้นที่ร้านที่ไกลที่สุด ของเส้นทางรถ นั้น

ระยะทางรวม(ก.ม.)		203,637		การคำนวณระยะทาง		Swap	
ค่าขนส่งรวม(บาท)		304,402		การคำนวณรถ		Full Truck Load	
เวลาที่ใช้คำนวณ(วินาที)		3,804					

ลำดับที่.	รอบที่	หมายเลขรถ	ชนิดของรถ	ความจุของรถ	ชื่อร้าน	ความจุที่ออเดอร์	ความจุรวม	ระยะทาง	ค่าขนส่ง	ค่าขนส่งที่คิด
1	1	1	6W	10	AGI	4	10	245.28	1,264.00	1,373.00
2					JH	2		71.08	1,373.00	
3					ET	3		8.60	1,373.00	
4					RH	1		19.77	1,373.00	
					DCKK			280.16		
					รวมระยะทาง			624.89		
5	1	2	6W	10	EX	3	10	201.91	980.25	1,027.75
6					ADH	3		44.14	980.25	
7					BF	4		338.74	1,027.75	
					DCKK			140.81		
					รวมระยะทาง			725.60		
8	1	3	6W	10	ZZ	2	10	307.87	1,367.25	1,367.25
9					ACU	2		32.44	1,367.25	
10					AFT	5		104.25	1,298.75	
11					ADM	1		39.89	1,298.75	
					DCKK			243.18		
					รวมระยะทาง			727.63		
12	1	4	6W	10	MI	4	10	118.91	783.00	821.75
13					QJ	3		116.22	821.75	
14					W	3		0.12	821.75	
					DCKK			85.37		
					รวมระยะทาง			320.62		
15	1	5	6W	10	XF	4	10	250.80	1,068.50	1,068.50

รูปที่ 3.27 รูปแบบของการแสดงผลของ Export File รูปที่ 1

ซีทที่ 2 (ชื่อซีท ระยะทาง) แสดงลำดับร้านทั้งหมดที่ได้ก่อนการปรับปรุง

เส้นทาง

No.	Shop Name	Distance	Total Order
1	DCKK	0	0
2	AGI	245.28	4
3	JH	71.08	2
4	ET	8.6	3
5	EX	367.27	3
6	ADH	44.14	3
7	BF	338.74	4
8	ZZ	387.74	2
9	ACU	32.44	2
10	RH	89.25	1
11	AFT	177.87	5
12	MI	360.11	4
13	QJ	116.22	3
14	XE	321.26	4
15	HR	40.92	4
16	W	290.31	3
17	SH	192.98	2
18	WK	331.12	4
19	EC	9.46	4
20	AFO	58.73	6
21	EB	48.1	4
22	GE	351.92	2
23	AFE	239.17	3
24	MG	25.79	2
25	QG	22.67	5
26	ADM	175.73	1

รูปที่ 3.28 รูปแบบของการแสดงผลของ Export File รูปที่ 2

บทที่ 4

ผลการวิจัยและการวิเคราะห์ผลการทดลอง

4.1 ผลการวิจัย

เปรียบเทียบค่าขนส่งและเวลาการคำนวณ ที่เกิดจากการปรับปรุงเส้นทางทั้งหมด 9 วิธี
กับการไม่ปรับปรุงเส้นทาง (Normal)

4.1.1 แสดงผลค่าขนส่งและเวลาการคำนวณ ของการบรรทุกสินค้าแบบ FTL ที่มีระดับ
ยอดสั่งสินค้าสูง

ตารางที่ 4.1 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดสั่งสินค้าสูง และมีการบรรทุกแบบ FTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	255,390	255,390	255,390	255,390	255,390
S	230,765	231,557	230,118	248,022	230,118
I	246,953	247,880	247,260	246,023	246,023
R	245,388	246,005	246,794	246,774	245,388
S-I	234,853	235,073	234,482	233,939	233,939
I-S	234,774	234,203	235,402	233,872	233,872
S-R	233,612	234,894	233,428	235,584	233,428
R-S	233,137	234,453	232,662	234,190	232,662
I-R	247,661	246,730	245,183	246,097	245,183
R-I	244,525	244,664	246,118	244,639	244,525

จากตารางที่ 4.1 พบว่าการปรับปรุงเส้นทางด้วยวิธี SWAP และ REVERSE-SWAP มีค่า
ขนส่งต่ำที่สุดตามลำดับ

ตารางที่ 4.2 เปรียบเทียบเวลาการคำนวณ (นาทื) ของระดับยอดสั่งสินค้าสูง บรรทุกแบบ FTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	48	48	48	48	48
S	28	21	23	23	21
I	28	26	26	25	25
R	27	24	25	25	24
S-I	23	23	23	21	21
I-S	25	24	25	23	23
S-R	24	25	25	22	22
R-S	24	25	25	23	23
I-R	25	25	29	24	24
R-I	28	26	25	25	25

จากตารางที่ 4.2 แสดงเวลาการคำนวณ การปรับปรุงเส้นทางแบบ SWAP และ REVERSE-SWAP ใช้เวลาการคำนวณ 21 และ 23 นาทื ตามลำดับ

4.1.2 แสดงผลค่าขนส่งและเวลาการคำนวณ ของการบรรทุกสินค้าแบบ FTL ที่มีระดับยอดสั่งสินค้าปานกลาง

ตารางที่ 4.3 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดสั่งสินค้าปานกลาง และมีการบรรทุกแบบ FTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	238,832	238,832	238,832	238,832	238,832
S	204,576	204,368	204,313	205,540	204,313
I	221,863	222,546	221,840	223,513	221,840
R	219,670	219,925	220,164	221,110	219,670
S-I	208,464	210,020	210,009	208,146	208,146
I-S	209,823	208,489	208,989	209,132	208,489
S-R	207,358	206,863	209,732	208,539	206,863

ตารางที่ 4.3 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดสั่งสินค้าปานกลาง และมีการบรรจุทุกแบบ FTL (ต่อ)

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
R-S	206,535	208,101	206,651	207,588	206,535
I-R	221,699	221,061	222,125	219,877	219,877
R-I	216,506	222,035	219,287	220,574	216,506

จากตารางที่ 4.3 พบว่าการปรับปรุงเส้นทางด้วยวิธี SWAP และ REVERSE-SWAP มีค่าขนส่งต่ำที่สุด ตามลำดับ

ตารางที่ 4.4 เปรียบเทียบเวลาการคำนวณ (นาทื) ของระดับยอดสั่งสินค้าปานกลาง และมีการบรรจุทุกแบบ FTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	48	48	48	48	48
S	22	21	26	21	21
I	30	25	27	24	24
R	28	28	26	24	24
S-I	21	22	22	21	21
I-S	27	23	23	23	23
S-R	27	26	27	22	22
R-S	24	24	27	24	24
I-R	26	28	26	25	25
R-I	27	28	27	27	27

จากตารางที่ 4.4 แสดงเวลาการคำนวณ การปรับปรุงเส้นทางแบบ SWAP และ REVERSE-SWAP ใช้เวลาการคำนวณ 21 และ 24 นาทื ตามลำดับ

4.1.3 แสดงผลค่าขนส่งและเวลาการคำนวณ ของการบรรจุทุกสินค้าแบบ FTL ที่มีระดับยอดสั่งซื้อสินค้าต่ำ

ตารางที่ 4.5 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดสั่งซื้อสินค้าต่ำ และมีการบรรจุทุกแบบ FTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	163,984	163,984	163,984	163,984	163,984
S	166,747	167,617	166,214	167,400	166,214
I	180,383	180,517	180,128	180,385	180,128
R	176,589	178,095	178,801	179,078	176,589
S-I	169,910	171,739	170,840	169,809	169,809
I-S	170,362	169,838	169,395	170,732	169,395
S-R	169,236	171,460	169,450	169,711	169,236
R-S	169,361	170,387	169,020	167,790	167,790
I-R	180,834	179,832	179,424	180,573	179,424
R-I	178,763	177,701	177,613	178,937	177,613

จากตารางที่ 4.5 พบว่าการไม่ปรับปรุงเส้นทาง (Normal) และ SWAP มีค่าขนส่งต่ำที่สุดตามลำดับ

ตารางที่ 4.6 เปรียบเทียบเวลาการคำนวณ (นาทีก) ของระดับยอดสั่งซื้อสินค้าต่ำ และมีการบรรจุทุกแบบ FTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	48	48	48	48	48
S	21	22	21	26	21
I	23	25	25	25	23
R	25	25	27	26	25
S-I	22	22	22	25	22
I-S	23	23	24	24	23
S-R	22	22	20	30	20

ตารางที่ 4.6 เปรียบเทียบเวลาการคำนวณ (นาทื) ของระดับยอดสั่งสินค้าต่ำ และมีการบรรทุกแบบ FTL (ต่อ)

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
R-S	25	22	25	24	22
I-R	25	26	25	25	25
R-I	25	26	26	26	25

จากตารางที่ 4.6 แสดงเวลาการคำนวณ แบบไม่ปรับปรุงเส้นทาง (Normal) และการปรับปรุงเส้นทางแบบ SWAP ใช้เวลาการคำนวณ 48 และ 21 นาทื ตามลำดับ สรุปผลวิจัยที่ได้ด้านค่าขนส่งจากตารางที่ 4.1, 4.3 และ 4.5 เป็นดังตารางที่ 4.7

ตารางที่ 4.7 สรุปผลวิจัยที่ได้ สำหรับกรณีการบรรทุกสินค้าแบบ Full Truck Load

ระดับการสั่งสินค้า	Algorithm ที่มีค่าขนส่งต่ำที่สุด 2 อันดับแรก ตามลำดับ
สูง	SWAP และ REVERSE-SWAP
ปานกลาง	SWAP และ REVERSE-SWAP
ต่ำ	Normal และ SWAP

จากสรุปผลการวิจัย กรณีระดับการสั่งสินค้าสูงและปานกลางพบว่า การปรับปรุงเส้นทางด้วยวิธี SWAP และ REVERSE-SWAP เป็นวิธีที่ได้ค่าขนส่งต่ำ และเมื่อทำการทดสอบ T test ของการปรับปรุงเส้นทางด้วยวิธี SWAP และ REVERSE-SWAP โดยตั้งสมมติฐานว่า ผลของค่าขนส่งที่เกิดจากการปรับปรุงเส้นทางด้วยวิธี SWAP และ REVERSE-SWAP ไม่แตกต่างกัน ที่ระดับความเชื่อมั่น 0.10 พบว่าค่า p-value เท่ากับ 0.406 แสดงว่าผลของค่าขนส่งที่ได้จากการปรับปรุงเส้นทางแบบ SWAP และผลของค่าขนส่งที่ได้จากการปรับปรุงเส้นทางแบบ REVERSE-SWAP ไม่แตกต่างกัน ในระดับการสั่งสินค้าสูงและปานกลาง

สำหรับกรณีระดับการสั่งสินค้าต่ำ วิธีไม่ปรับปรุงเส้นทาง และปรับปรุงเส้นทางด้วยวิธี SWAP เป็นวิธีที่ได้ค่าขนส่งต่ำ และเมื่อทำการทดสอบ T test โดยตั้งสมมติฐานว่า ผลของค่าขนส่งที่เกิดจากการไม่ปรับปรุงเส้นทาง และ การปรับปรุงเส้นทางด้วยวิธี SWAP ไม่แตกต่างกัน ที่ระดับความเชื่อมั่น 0.10 พบว่าค่า p-value เท่ากับ 0.001 แสดงว่าผลของค่าขนส่งที่ได้จากการไม่ปรับปรุงเส้นทาง ดีกว่า การปรับปรุงเส้นทางแบบ SWAP

4.1.4 แสดงผลค่าขนส่งและเวลาการคำนวณ ของการบรรจุทุกสินค้าแบบ LTL ที่มีระดับยอดส่งสินค้าสูง

ตารางที่ 4.8 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดส่งสินค้าสูง และมีการบรรจุทุกแบบ LTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	255,390	255,390	255,390	255,390	255,390
S	236,506	235,323	236,344	234,511	234,511
I	249,683	252,034	252,879	250,850	249,683
R	257,451	256,687	259,253	256,564	256,564
S-I	234,971	236,380	235,734	238,926	234,971
I-S	236,503	236,229	239,344	237,263	236,229
S-R	238,991	240,172	238,796	240,739	238,796
R-S	239,073	238,667	238,593	238,059	238,059
I-R	249,469	250,823	250,453	250,581	249,469
R-I	249,863	249,442	253,360	252,257	249,442

จากตารางที่ 4.8 พบว่าการปรับปรุงเส้นทางด้วยวิธี SWAP และ SWAP-INSERT มีค่าขนส่งต่ำที่สุด ตามลำดับ

ตารางที่ 4.9 เปรียบเทียบเวลาการคำนวณ (นาทีก) ของระดับยอดส่งสินค้าสูง และมีการบรรจุทุกแบบ LTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	48	48	48	48	48
S	18	19	20	20	18
I	21	20	21	22	20
R	22	22	24	25	22
S-I	20	21	21	24	20
I-S	21	25	21	20	20
S-R	20	19	21	21	19

ตารางที่ 4.9 เปรียบเทียบเวลาการคำนวณ (นาทื) ของระดับยอดสั่งสินค้าสูง และมีการบรรจุทุกแบบ LTL (ต่อ)

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
R-S	21	22	23	22	21
I-R	21	21	23	21	21
R-I	22	22	21	21	21

จากตารางที่ 4.9 แสดงเวลาการคำนวณ การปรับปรุงเส้นทางแบบ SWAP และ SWAP-INSERT ใช้เวลาการคำนวณ 18 และ 20 นาทื ตามลำดับ

4.1.5 แสดงผลค่าขนส่งและเวลาการคำนวณ ของการบรรจุทุกสินค้าแบบ LTL ที่มีระดับยอดสั่งสินค้าปานกลาง

ตารางที่ 4.10 เปรียบเทียบเวลาการคำนวณ (นาทื) ของระดับยอดสั่งสินค้าสูง และมีการบรรจุทุกแบบ LTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	238,832	238,832	238,832	238,832	238,832
S	211,722	208,525	211,998	210,131	208,525
I	222,018	221,375	222,409	221,081	221,081
R	226,077	223,399	227,947	229,226	223,399
S-I	211,885	212,917	210,638	211,706	210,638
I-S	212,008	215,902	212,934	212,491	212,008
S-R	210,799	212,863	211,990	210,642	210,642
R-S	212,408	211,920	211,508	212,727	211,508
I-R	221,452	220,237	224,727	226,545	220,237
R-I	224,631	220,944	223,627	223,847	220,944

จากตารางที่ 4.10 พบว่าการปรับปรุงเส้นทางด้วยวิธี SWAP และ SWAP-INSERT มีค่าขนส่งต่ำที่สุด ตามลำดับ

ตารางที่ 4.11 เปรียบเทียบเวลาการคำนวณ (นาทื) ของระดับยอตสังสินค้ำปานกลาง และมีการ
บรรทุกแบบ LTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	48	48	48	48	48
S	18	20	20	21	18
I	23	24	22	22	22
R	22	23	23	26	22
S-I	20	21	20	21	20
I-S	21	22	21	22	21
S-R	26	20	21	20	20
R-S	23	22	18	22	18
I-R	22	23	21	21	21
R-I	26	25	27	26	25

จากตารางที่ 4.11 แสดงเวลาการคำนวณ การปรับปรุงเส้นทางแบบ SWAP และ SWAP-INSERT ใช้เวลาการคำนวณ 18 และ 20 นาทื ตามลำดับ

4.1.6 แสดงผลค่าขนส่งและเวลาการคำนวณ ของการบรรทุกสินค้ำแบบ LTL ที่มีระดับ ยอตสังสินค้ำต่ำ

ตารางที่ 4.12 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอตสังสินค้ำปริมาณต่ำ และมีการบรรทุกแบบ
LTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	163,984	163,984	163,984	163,984	163,984
S	172,085	173,959	172,063	169,328	169,328
I	183,735	184,267	183,486	184,846	183,486
R	187,550	183,892	185,833	185,582	183,892
S-I	172,129	169,274	173,278	174,098	169,274
I-S	174,441	173,593	175,206	172,548	172,548
S-R	173,877	173,706	177,081	172,681	172,681

ตารางที่ 4.12 เปรียบเทียบค่าขนส่ง (บาท) ของระดับยอดสั่งสินค้าปริมาณต่ำ และมีการบรรทุกแบบ LTL (ต่อ)

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
R-S	172,312	173,362	175,171	172,954	172,312
I-R	185,967	183,945	185,864	182,916	182,916
R-I	182,705	180,952	182,141	182,601	180,952

จากตารางที่ 4.12 พบว่าการไม่ปรับปรุงเส้นทาง และการปรับปรุงเส้นทางแบบ SWAP-INSERT มีค่าขนส่งต่ำที่สุด ตามลำดับ

ตารางที่ 4.13 เปรียบเทียบเวลาการคำนวณ (นาทื) ของระดับยอดสั่งสินค้าต่ำ และมีการบรรทุกแบบ LTL

Algorithm	ผลครั้งที่ 1	ผลครั้งที่ 2	ผลครั้งที่ 3	ผลครั้งที่ 4	ค่าที่ดีที่สุด
Normal	48	48	48	48	48
S	22	18	23	23	18
I	22	24	23	21	21
R	28	23	21	24	21
S-I	20	21	21	20	20
I-S	19	24	21	24	19
S-R	20	22	22	20	20
R-S	20	20	24	21	20
I-R	19	21	21	22	19
R-I	18	21	21	21	18

จากตารางที่ 4.13 แสดงเวลาการคำนวณ แบบไม่ปรับปรุงเส้นทาง (Normal) และการปรับปรุงเส้นทางแบบ SWAP-INSERT ใช้เวลาการคำนวณ 48 และ 20 นาที ตามลำดับ สรุปผลวิจัยที่ได้ สำหรับกรณีการบรรทุกสินค้าแบบ Less than Truck Load เป็นดังตารางที่ 4.14

สรุปผลวิจัยที่ได้ สำหรับกรณีการบรรทุกสินค้าแบบ Less than Truck Load เป็นดังตารางที่ 4.14

ตารางที่ 4.14 สรุปผลวิจัยที่ได้ สำหรับกรณีการบรรทุกสินค้าแบบ Less than Truck Load

ระดับการสั่งสินค้า	Algorithm ที่มีค่าขนส่งต่ำที่สุด 2 อันดับแรก ตามลำดับ
สูง	SWAP และ SWAP-INSERT
ปานกลาง	SWAP และ SWAP-INSERT
ต่ำ	Normal และ SWAP-INSERT

จากสรุปผลการวิจัย กรณีระดับการสั่งสินค้าสูงและปานกลางพบว่า การปรับปรุงเส้นทางด้วยวิธี SWAP และ SWAP-INSERT เป็นวิธีที่ได้ค่าขนส่งต่ำ และเมื่อทำการทดสอบ T test ของการปรับปรุงเส้นทางด้วยวิธี SWAP และ SWAP-INSERT โดยตั้งสมมติฐานว่า ผลของค่าขนส่งที่เกิดจากการปรับปรุงเส้นทางด้วยวิธี SWAP และ SWAP-INSERT ไม่แตกต่างกัน ที่ระดับความเชื่อมั่น 0.10 พบว่าค่า p-value เท่ากับ 0.132 แสดงว่าผลของค่าขนส่งที่ได้จากการปรับปรุงเส้นทางแบบ SWAP และ ผลของค่าขนส่งที่ได้จากการปรับปรุงเส้นทางแบบ SWAP-INSERT ไม่แตกต่างกัน ในระดับการสั่งสินค้าสูงและปานกลาง

สำหรับกรณีระดับการสั่งสินค้าต่ำ วิธีไม่ปรับปรุงเส้นทาง และปรับปรุงเส้นทางด้วยวิธี SWAP-INSERT เป็นวิธีที่ได้ค่าขนส่งต่ำ และเมื่อทำการทดสอบ T test โดยตั้งสมมติฐานว่า ผลของค่าขนส่งที่เกิดจากการไม่ปรับปรุงเส้นทาง และ การปรับปรุงเส้นทางด้วยวิธี SWAP-INSERT ไม่แตกต่างกัน ที่ระดับความเชื่อมั่น 0.10 พบว่าค่า p-value เท่ากับ 0.002 แสดงว่าผลของค่าขนส่งที่ได้จากการไม่ปรับปรุงเส้นทาง ดีกว่า การปรับปรุงเส้นทางแบบ SWAP

4.2 วิเคราะห์ผล

จากสรุปผลการวิจัย พบว่า การขนส่งแบบ Full Truck Load กรณีระดับการสั่งสินค้าสูงและปานกลาง วิธีการปรับปรุงเส้นทางที่เหมาะสม เพื่อให้มีค่าขนส่งต่ำที่สุดสามารถใช้ได้ทั้งวิธีการ SWAP หรือ REVERSE-SWAP ส่วนของการขนส่งแบบ Less than Truck Load การปรับปรุงเส้นทางด้วยวิธี SWAP และ SWAP-INSERT เป็นวิธีที่เหมาะสมที่สุด

สำหรับกรณีระดับการสั่งสินค้าต่ำ ทั้งการขนส่งแบบ Full Truck Load และ Less than Truck Load พบว่าเป็นวิธีที่เหมาะสมที่สุด แต่หากดูด้านเวลาการประมวลผลจะพบว่าเวลาการคำนวณสูงกว่าประมาณ 50% ซึ่งในการดำเนินการหากในกรณีรีบเร่งอาจใช้การคำนวณแบบปรับปรุงเส้นทางในการดำเนินการแทน แต่ค่าขนส่งจะสูงขึ้นประมาณ 1-3%

บทที่ 5

สรุปผลการวิจัย

5.1 สรุปผลการวิจัย

สำหรับการบรรทุกสินค้าแบบ Full Truck Load วิธีปรับปรุงเส้นทางที่เหมาะสมสำหรับระดับการส่งสินค้าสูงและปานกลาง คือ SWAP และ REVERSE-SWAP สำหรับ ระดับการส่งสินค้าต่ำ คือ การไม่ปรับปรุงเส้นทาง

สำหรับการบรรทุกสินค้าแบบ Less than Truck Load วิธีปรับปรุงเส้นทางที่เหมาะสมสำหรับ ระดับการส่งสินค้าสูงและปานกลาง คือ SWAP และ SWAP-INSERT สำหรับ ระดับการส่งสินค้าต่ำ คือ การไม่ปรับปรุงเส้นทาง

ทั้งนี้ กรณีการส่งสินค้าแบบ Full Truck Load เป็นการขนส่งสินค้าที่ค่าขนส่งต่ำกว่าการส่งสินค้าแบบ Less than Truck Load

5.2 ข้อเสนอแนะ

จากงานวิจัยเป็นการปรับปรุงเส้นทาง 2 ขั้นตอน ซึ่งอาจมีค่าขนส่งที่ต่ำกว่านี้หากมีการปรับปรุงเส้นทางหลายขั้นตอนมากขึ้น และเพิ่มการวิเคราะห์เปรียบเทียบการบรรทุกสินค้าแบบ FTL และ LTL



บรรณานุกรม

บรรณานุกรม

- กวี ศรีเมือง. (2550). การหาจำนวนพาหนะบรรทุกที่เหมาะสมในการขนส่งสินค้าในธุรกิจค้าปลีก
กรณีศึกษาที่อปส์ซูเปอร์มาร์เก็ต. วิทยานิพนธ์ วศ.ม. (วิศวกรรมโยธา). กรุงเทพฯ :
 คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.
- โกศล ดีศีลธรรม. (2551). โลจิสติกส์และห่วงโซ่อุปทานสำหรับการแข่งขันยุคใหม่. กรุงเทพฯ :
 บริษัท ฐานการพิมพ์ จำกัด.
- ชัยยศ ผิวปานแก้ว; และอนันต์ มุ่งวัฒนา. (2556). การพัฒนาอัลกอริทึมเพื่อกำหนดวิธีการการ
จัดเส้นทางรถขนส่งสินค้าให้มีค่าใช้จ่ายการขนส่งต่ำที่สุด. วิทยานิพนธ์ วศ.ม. (วิศวกรรม
 อุตสาหการ). กรุงเทพฯ : คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.
- ชัยยา นุรักษ์เช. (2548). การจัดเส้นทางเดินรถขนส่งขนมในเขตกรุงเทพฯและปริมณฑล
กรณีศึกษา : ห้างไอ.ซี.ซี.ฟฟลาย. วิทยานิพนธ์ วศ.ม. (วิศวกรรมอุตสาหการ). กรุงเทพฯ
 : คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.
- ชูลีกร ชนะสิทธิ์; และสรวิชัย เยาวสุวรรณ์ไชย. (2556 มกราคม-เมษายน). การพัฒนาโปรแกรมการ
 จัดเส้นทางรถขนส่งสินค้า กรณีศึกษาบริษัทผู้ให้บริการด้านธุรกิจขนส่งสินค้า.
WMS Journal of Management. ม.ป.ป.(2) : 55-69.
- ฐิตินันท์ ศรีสุวรรณดี; และระพีพันธ์ ปิตาคะโส. (2555). การแก้ปัญหาการจัดเส้นทางรถขนส่ง
ยานพาหนะด้วยวิธีการอาณานิคมมด กรณีศึกษา บริษัท เจียรนัยน้ำดื่ม จำกัด.
 วิทยานิพนธ์ วศ.ม. (วิศวกรรมอุตสาหการ). กรุงเทพฯ : คณะวิศวกรรมศาสตร์
 มหาวิทยาลัยอุบลราชธานี.
- ณกร อินทร์พยุ. (2548). การแก้ปัญหาการตัดสินใจในอุตสาหกรรมรถขนส่งและลอจิสติกส์
(Discrete Optimization in Transport and Logistics). กรุงเทพฯ : บริษัท ซีเอ็ด
 ยูเคชั่น จำกัด (มหาชน).
- ปาสีรัฐ บุญก่อน. (2554). แนวทางการพัฒนาการจัดเส้นทางเดินรถขนส่งสินค้า กรณีศึกษา ศูนย์
กระจายสินค้าประเภทเซรามิค. วิทยานิพนธ์ วท.ม. (การจัดการโลจิสติกส์). กรุงเทพฯ :
 บัณฑิตวิทยาลัย มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี.
- ระพีพันธ์ ปิตาคะโส. (2554). วิธีการเมตาฮิวริสติก Metaheuristic เพื่อแก้ไขปัญหา
การวางแผนการผลิตและการจัดการโลจิสติกส์. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี
 (ไทย-ญี่ปุ่น).

อุบลรัตน์ เจริญนาคม. (2551). การใช้วิธีฮิวริสติกเพื่อแก้ปัญหาการจัดเส้นทางยานพาหนะที่มี
คลังสินค้าหลายแห่ง. วิทยานิพนธ์ วท.ม. (วิศวกรรมอุตสาหการ). กรุงเทพฯ :
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์.

Asvin Goel; and Volker Gruhn. (2008). **A General Vehicle Routing Problem.**
European Journal of Operational Research. 191 (3) : 650-660.

Clarke G; and Wright J. (1963). Scheduling of Vehicles from a Central Depot to a
Number Delivery Points. **Operational Research.** 11 : 568-581.

Claudio B; Cunha; and Marcos Roberto Silva. (2005, March). A genetic Algorithm for
the Problem of Configuring a Hub-and-Spoke Network for a LTL Trucking
Company in Brazil. **European Journal of Operational Research.** 179 (3) :
747-758.

Dantzing G.B.; and Ramser J.H. (1959, October). The Truck Dispatching Problem.
Management Science 6. 6 (1) : 80-91.

Golden B; T. Magnanti; and H. Nguyen. (1977). Implementing Vehicle Routing
Algorithms. **Network an International Journal.** 7 (2) : 113-148.

Jean-Francois Cote; and Jean-Yves Potvin. (2009, October). A Tabu Search Heuristic
for the Vehicle Routing Problem with Private Fleet and Common Carrier.
European Journal of Operational Research. 198 (2) : 464-469.

Jing Fan. (2011). The Vehicle Routing Problem with Simultaneous Pickup and
Delivery Based on Customer Satisfaction. **Procedia Engineering.**
15 : 5,284-5,289.

Jose Brandao. (2007, May). A Deterministic Tabu Search Algorithm for the Fleet Size
and Mix Vehicle Routing Problem. **European Journal of Operational
Research.** 195 (3) : 716-728.

L.Bodin; et al. (1983). Routing and Scheduling of Vehicles and Crews. **Computer
and Operation Research.** 10(2) : 63-211.

Milan Stanojevic; Bogdana Stanojevic; and Mirko Vujošević. (2013, June). Enhances
Saving Calculation and its Applications for Solving Capacitated Vehicle
Routing Problem. **Applies Mathmatics and Computation.** 219 (20) :
10,302-10,312.

- Mostepha R; et al. (2011, November). A Comparative Study Between Dynamic Adapted PSO and VNS for the Vehicle Routing Problem with Dynamic Requests. **Applied Soft Computing**. 12 (4) : 1,426-1,439.
- Prins C. (2004). A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem. **Computer & Operational Research**. 31 (12). : 1,985-2,002.
- Ran Liu; Xiaolan XIE; and Vincent Augusto Carlos Rodriguez. (2013). Heuristic Algorithms for a Vehicle Routing Problem with Simultaneous Delivery and Pick up and Time Window in Home Health Care. **European Journal of Operational Research**. 230 (3) : 475-486.
- Thanchuda Phannikul; and Sombat Sindhuchao. (2010). **A Customized Tabu Search for the Vehicle Routing Problem with Simultaneous Pick-Up and Delivery**. Thesis M.Eng. (Industrial Engineering). Ubonratchathani : Department of Industrial Engineering, Ubonratchathani University.



ภาคผนวก

ภาคผนวก ก.
Coding

TNI

```

using Microsoft.Office.Interop.Excel;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Globalization;
using System.IO;
using System.Text;
using System.Threading;
using System.Windows.Forms;
using System.Configuration;
using System.Data.OleDb;

namespace PathRouting
{
    public partial class frmRouting : Form
    {
        private const int CP_NOCLOSE_BUTTON = 0x200;
        protected override CreateParams CreateParams
        {
            public frmRouting()
            {
                private class Shop
                private class Order
                private class TruckDetails
                private class calResult

                int[] LastSequence;
                int[] Sequence;
                int OrderNum = 0;
                bool flagFileLoad;

```

```

Shop shop = new Shop();
Order order = new Order();
TruckDetails truck = new TruckDetails();

private void frmRouting_Load(object sender, EventArgs e)
private void btnBrowseAllDistance_Click(object sender, EventArgs e)
{

    string strFile = browseCsvFile();
    if (strFile.ToUpper().EndsWith(".CSV"))
    {
        this.lblAllDistancePath.Text = strFile;
        putConfig("AllDistancePath", strFile);
    }
}

private void btnBrowseOrderPath_Click(object sender, EventArgs e)
{
    string strFile = browseXlsFile();
    if (strFile.ToUpper().EndsWith(".XLS") || strFile.ToUpper().EndsWith(".XLSX"))
    {
        this.lblOrderPath.Text = strFile;
        putConfig("OrderPath", strFile);
    }
}

private void btnBrowseTransport_Click(object sender, EventArgs e)
{
    string strFile = browseXlsFile();
    if (strFile.ToUpper().EndsWith(".XLS") || strFile.ToUpper().EndsWith(".XLSX"))
    {
        this.lblTransportFeePath.Text = strFile;
        putConfig("TransportFeePath", strFile);
    }
}

```

```

    }
}

private void btnRoute_Click(object sender, EventArgs e)
{
    if (!flagFileLoad)
    {
        MessageBox.Show("ยังไม่ได้โหลดไฟล์ กรุณากดปุ่ม Load File ก่อน");
        return;
    }
    int[] Sequence2;
    EnableAllButton(false);

    this.dataGridView1.Rows.Clear();
    this.dataGridView1.Columns.Clear();
    this.lblTotalDistance.Text = "";
    this.lblTotalTransportFee.Text = "";
    this.lblSpendTime.Text = "";
    this.progressBar1.Value = 0;
    this.lblProgress.Text = "0/100 %";
    this.lblStatus.Text = "กำลังโหลดข้อมูลจากไฟล์ระยะทาง";
    System.Windows.Forms.Application.DoEvents();

    int calMethod;
    if (this.radioFTL.Checked)
        calMethod = 1;
    else
        calMethod = 2;

    Sequence = new int[LastSequence.Length];
    Sequence = LastSequence;

```

```

        double TransportFee = TruckAndTransportFeeCal2(Sequence, shop, order,
truck, calMethod);

        bool isNewSequence = false;

        int divider = 0;
        if (this.textBox1.Text != "") divider++;
        if (this.textBox2.Text != "") divider++;
        if (this.textBox3.Text != "") divider++;

        switch (divider)
        {
            case 0:
                divider = 1;
                break;
            case 1:
                divider = 2;
                break;
            case 2:
                divider = 4;
                break;
            case 3:
                divider = 6;
                break;
        }
        this.progressBar1.Minimum = 0;
        this.progressBar1.Maximum = (OrderNum * (OrderNum - 1) / divider) - 1;
        DateTime beginTime = System.DateTime.Now;

        for (int j = 1; j <= 3; j++)
        {
            switch (this.panel1.Controls["textBox" + j].Text.ToLower())

```

```

{
    case "insert":
        this.lblStatus.Text = "กำลังคำนวณแบบ Insert";
        break;
    case "reverse":
        this.lblStatus.Text = "กำลังคำนวณแบบ Reverse";
        break;
    case "swap":
        this.lblStatus.Text = "กำลังคำนวณแบบ Swap";
        break;
    default:
        continue;
}
for (int i = 0; i < (OrderNum * (OrderNum - 1) / divider); i++)
{
    Sequence2 = new int[order.ShopName.Count];
    switch (this.panel1.Controls["textBox" + j].Text.ToLower())
    {
        case "insert":
            Sequence2 = newSequenceFromInsert(order.Distance, Sequence,
            out isNewSequence);
            break;
        case "reverse":
            Sequence2 = newSequenceFromReverse(order.Distance,
            Sequence, out isNewSequence);
            break;
        case "swap":
            Sequence2 = newSequenceFromSwap(order.Distance, Sequence,
            out isNewSequence);
            break;
    }
}

```

```

        if (isNewSequence)
        {
            double newTransportFee = TruckAndTransportFeeCal2(Sequence2,
shop, order, truck, calMethod);
            if (newTransportFee < TransportFee)
            {
                TransportFee = newTransportFee;
                Sequence = Sequence2;
            }
        }
        this.progressBar1.Value = i;
        this.lblProgress.Text = ((double)i++ / ((OrderNum * (OrderNum - 1) /
divider) - 1) * 100).ToString("0.0") + "/100 %";
        System.Windows.Forms.Application.DoEvents();
    }
}

calResult result = TruckAndTransportFeeCal(Sequence, shop, order, truck,
calMethod);

DateTime endTime = System.DateTime.Now;

this.dataGridView2.Columns.Clear();
this.dataGridView2.Columns.Add("Col1", "No.");
this.dataGridView2.Columns.Add("Col2", "Shop Name");
this.dataGridView2.Columns.Add("Col3", "Distance");
this.dataGridView2.Columns.Add("Col4", "Total Order");
this.dataGridView2.Rows.Add(new string[] { "1",order.ShopName[Sequence[0]],
order.Distance[Sequence[0],Sequence[0]].ToString(),order.Volume[Sequence[0]].ToStri
ng()});

for (int i = 1; i < Sequence.Length; i++)
{

```



```

        this.dataGridView2.Rows.Add(new string[] { (i + 1).ToString(),
order.ShopName[Sequence[i]], order.Distance[Sequence[i - 1], Sequence[i]].ToString(),
order.Volume[Sequence[i]].ToString()});
    }

```

```

this.dataGridView1.Columns.Clear();
this.dataGridView1.Columns.Add("Col1", "ลำดับที่.");
this.dataGridView1.Columns.Add("Col2", "รอบที่");
this.dataGridView1.Columns.Add("Col3", "หมายเลขรถ");
this.dataGridView1.Columns.Add("Col4", "ชนิดของรถ");
this.dataGridView1.Columns.Add("Col5", "ความจุของรถ");
this.dataGridView1.Columns.Add("Col6", "ชื่อร้าน");
this.dataGridView1.Columns.Add("Col7", "ความจุที่ออเดอร์");
this.dataGridView1.Columns.Add("Col8", "ความจุรวม");
this.dataGridView1.Columns.Add("Col9", "ระยะทางจาก DC");
this.dataGridView1.Columns.Add("Col10", "ค่าขนส่ง");
this.dataGridView1.Columns.Add("Col11", "ค่าขนส่งที่คิด");

```

```

int ShopNo = 0;
var iRound = 0;
double TotalTransportFee = 0;
double TotalDistance = 0;
foreach (var t in result.Truck6Cal)
{
    int truckNo = 0;
    var truckFee = result.Truck6FeeCal.ToArray();
    var truckTotalVol = result.Truck6TotalVol.ToArray();
    foreach (int[] tr in t)
    {
        double subDistance = 0;
        if (tr.Length == 0) continue;

```

```

int lastShop = 0;
for (int i = 0; i < tr.Length; i++)
{
    ShopNo++;
    if (i == 0)
    {
        this.dataGridView1.Rows.Add(new string[] { ShopNo.ToString(),
(iRound + 1).ToString(), (truckNo + 1).ToString(), "6W",
truck.Truck6MaxVolume.ToString("0.00"), order.ShopName[tr[i]],
order.Volume[tr[i]].ToString("0.00"), truckTotalVol[iRound][truckNo].ToString("0.00"),
shop.Distance[0, tr[i]].ToString(), truck.Truck6Fee[tr[i]].ToString("0.00"),
truckFee[iRound][truckNo].ToString("0.00") });

        TotalTransportFee += truckFee[iRound][truckNo];
        subDistance += shop.Distance[0, tr[i]];
    }
    else
    {
        this.dataGridView1.Rows.Add(new string[] { ShopNo.ToString(), "",
"", "", "", order.ShopName[tr[i]], order.Volume[tr[i]].ToString("0.00"), "", shop.Distance[tr[i]
- 1], tr[i]].ToString(), truck.Truck6Fee[tr[i]].ToString("0.00"), "" });

        subDistance += shop.Distance[tr[i] - 1], tr[i]];
    }
    lastShop = i;
}

this.dataGridView1.Rows.Add(new string[] { "", "", "", "", "",
order.ShopName[0], "", "", shop.Distance[0, tr[lastShop]].ToString(), "", "" });

subDistance += shop.Distance[0, tr[lastShop]];

this.dataGridView1.Rows.Add(new string[] { "", "", "", "", "", "รวมระยะทาง",
"", "", subDistance.ToString(), "", "" });

```

```

        TotalDistance += subDistance;
        truckNo++;
    }

    truckNo = 0;
    truckFee = result.Truck4FeeCal.ToArray();
    truckTotalVol = result.Truck4TotalVol.ToArray();
    foreach (int[] tr in result.Truck4Cal[iRound])
    {
        double subDistance = 0;
        if (tr.Length == 0) continue;
        int lastShop = 0;
        for (int i = 0; i < tr.Length; i++)
        {
            ShopNo++;
            if (i == 0)
            {
                this.dataGridView1.Rows.Add(new string[] { ShopNo.ToString(),
                (iRound + 1).ToString(), (truckNo + 1).ToString(), "4W",
                truck.Truck4MaxVolume.ToString("0.00"), order.ShopName[tr[i]],
                order.Volume[tr[i]].ToString("0.00"), truckTotalVol[iRound][truckNo].ToString("0.00"),
                shop.Distance[0, tr[i]].ToString(), truck.Truck4Fee[tr[i]].ToString("0.00"),
                truckFee[iRound][truckNo].ToString("0.00") });

                TotalTransportFee += truckFee[iRound][truckNo];
                subDistance += shop.Distance[0, tr[i]];
            }
            else
            {
                this.dataGridView1.Rows.Add(new string[] { ShopNo.ToString(), "",
                "", "", order.ShopName[tr[i]], order.Volume[tr[i]].ToString("0.00"), "", shop.Distance[tr[i]
                - 1, tr[i]].ToString(), truck.Truck4Fee[tr[i]].ToString("0.00"), "" });
            }
        }
    }

```

```

        subDistance += shop.Distance[tr[i - 1], tr[i]];
    }
    lastShop = i;
}
this.dataGridView1.Rows.Add(new string[] { "", "", "", "", "",
order.ShopName[0], "", "", shop.Distance[0, tr[lastShop]].ToString(), "", "" });
subDistance += shop.Distance[0, tr[lastShop]];
this.dataGridView1.Rows.Add(new string[] { "", "", "", "", "", "รวมระยะทาง",
"", "", subDistance.ToString(), "", "" });

TotalDistance += subDistance;
truckNo++;
}
iRound++;
}

dataGridView1.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
this.lblTotalDistance.Text = TotalDistance.ToString();
this.lblTotalTransportFee.Text = TotalTransportFee.ToString();
this.lblSpendTime.Text = (endTime -
beginTime).TotalMinutes.ToString("0.00");
this.lblStatus.Text = "คำนวณเสร็จเรียบร้อยแล้ว";
System.Windows.Forms.Application.DoEvents();

MessageBox.Show("คำนวณเสร็จเรียบร้อยแล้ว", "สถานะ", MessageBoxButtons.OK,
MessageBoxIcon.Information);
EnableAllButton();
}

```

```

private void btnExportToExcel_Click(object sender, EventArgs e)
private double computeTotalDistance(double[,] dblDistance, int[] Sequence)
private int[] newSequenceFromRandom(int ShopNum)
private int[] newSequenceFromSwap(double[,] dblDistance, int[] Sequence, out bool
IsNewSequence)
{
    int random1, random2;
    int[] Sequence2 = new int[Sequence.Length];
    Sequence.CopyTo(Sequence2, 0);
    Random rnd = new Random();
    do
    {
        random1 = rnd.Next(2, Sequence.Length + 1) - 1;
        random2 = rnd.Next(2, Sequence.Length + 1) - 1;
    } while (random1 == random2);
    Sequence2[random1] = Sequence[random2];
    Sequence2[random2] = Sequence[random1];

    if (computeTotalDistance(dblDistance, Sequence) <=
computeTotalDistance(dblDistance, Sequence2))
    {
        IsNewSequence = false;
        return Sequence;
    }
    else
    {
        IsNewSequence = true;
        return Sequence2;
    }
}

```

```
private int[] newSequenceFromSwap(double[,] dblDistance, int[] Sequence, out
double calDistance)
```

```
{
    int random1, random2;
    int[] Sequence2 = new int[Sequence.Length];
    Sequence.CopyTo(Sequence2, 0);
    Random rnd = new Random();
    do
    {
        random1 = rnd.Next(2, Sequence.Length + 1) - 1;
        random2 = rnd.Next(2, Sequence.Length + 1) - 1;
    } while (random1 == random2);
    Sequence2[random1] = Sequence[random2];
    Sequence2[random2] = Sequence[random1];

    double dist1 = computeTotalDistance(dblDistance, Sequence);
    double dist2 = computeTotalDistance(dblDistance, Sequence2);
    if (dist1 <= dist2)
    {
        calDistance = dist1;
        return Sequence;
    }
    else
    {
        calDistance = dist2;
        return Sequence2;
    }
}
```

```
private int[] newSequenceFromSwap(double[,] dblDistance, int[] Sequence, int
pos1, int pos2)
```

```

{
    int[] Sequence2 = new int[Sequence.Length];
    Sequence.CopyTo(Sequence2, 0);
    Sequence2[pos1] = Sequence[pos2];
    Sequence2[pos2] = Sequence[pos1];

    if (computeTotalDistance(dblDistance, Sequence) <=
computeTotalDistance(dblDistance, Sequence2))
    {
        return Sequence;
    }
    else
    {
        return Sequence2;
    }
}

private int[] newSequenceFromInsert(double[,] dblDistance, int[] Sequence, out
bool IsNewSequence)
{
    int random1, random2;
    int[] Sequence2 = new int[Sequence.Length];
    Sequence.CopyTo(Sequence2, 0);
    Random rnd = new Random();
    do
    {
        random1 = rnd.Next(2, Sequence.Length + 1) - 1;
        random2 = rnd.Next(2, Sequence.Length + 1) - 1;
    } while (random1 == random2);

    Sequence2[random2] = Sequence[random1];
    if (random1 > random2)

```



```

{
    for (int i = random1; i > random2; i--)
    {
        Sequence2[i] = Sequence[i - 1];
    }
}
else
{
    for (int i = random1; i < random2; i++)
    {
        Sequence2[i] = Sequence[i + 1];
    }
}
if (computeTotalDistance(dblDistance, Sequence) <=
computeTotalDistance(dblDistance, Sequence2))
{
    IsNewSequence = false;
    return Sequence;
}
else
{
    IsNewSequence = true;
    return Sequence2;
}
}

private int[] newSequenceFromInsert(double[,] dblDistance, int[] Sequence, out
double calDistance)
{
    int random1, random2;
    int[] Sequence2 = new int[Sequence.Length];
    Sequence.CopyTo(Sequence2, 0);

```

```
Random rnd = new Random();
do
{
    random1 = rnd.Next(2, Sequence.Length + 1) - 1;
    random2 = rnd.Next(2, Sequence.Length + 1) - 1;
} while (random1 == random2);
Sequence2[random2] = Sequence[random1];
if (random1 > random2)
{
    for (int i = random1; i > random2; i--)
    {
        Sequence2[i] = Sequence[i - 1];
    }
}
else
{
    for (int i = random1; i < random2; i++)
    {
        Sequence2[i] = Sequence[i + 1];
    }
}
double dist1 = computeTotalDistance(dblDistance, Sequence);
double dist2 = computeTotalDistance(dblDistance, Sequence2);
if (dist1 <= dist2)
{
    calDistance = dist1;
    return Sequence;
}
else
{
    calDistance = dist2;
```

```
        return Sequence2;
    }
}

private int[] newSequenceFromInsert(double[,] dblDistance, int[] Sequence, int
pos1, int pos2)
{
    int[] Sequence2 = new int[Sequence.Length];
    Sequence.CopyTo(Sequence2, 0);
    Sequence2[pos2] = Sequence[pos1];
    if (pos1 > pos2)
    {
        for (int i = pos1; i > pos2; i--)
        {
            Sequence2[i] = Sequence[i - 1];
        }
    }
    else
    {
        for (int i = pos1; i < pos2; i++)
        {
            Sequence2[i] = Sequence[i + 1];
        }
    }

    if (computeTotalDistance(dblDistance, Sequence) <=
computeTotalDistance(dblDistance, Sequence2))
    {
        return Sequence;
    }
    else

```

```
{
    return Sequence2;
}

private int[] newSequenceFromReverse(double[,] dblDistance, int[] Sequence,
out bool IsNewSequence)
{
    int random1, random2;
    int[] Sequence2 = new int[Sequence.Length];
    Sequence.CopyTo(Sequence2, 0);
    Random rnd = new Random();
    do
    {
        random1 = rnd.Next(2, Sequence.Length + 1) - 1;
        random2 = rnd.Next(2, Sequence.Length + 1) - 1;
    } while (random1 == random2);
    int j = random2;
    if (random1 > random2)
    {
        for (int i = random1; i >= random2; i--)
        {
            Sequence2[i] = Sequence[j++];
        }
    }
    else
    {
        for (int i = random1; i <= random2; i++)
        {
            Sequence2[i] = Sequence[j--];
        }
    }
}
```

```

    }
    if (computeTotalDistance(dblDistance, Sequence) <=
computeTotalDistance(dblDistance, Sequence2))
    {
        IsNewSequence = false;
        return Sequence;
    }
    else
    {
        IsNewSequence = true;
        return Sequence2;
    }
}

private int[] newSequenceFromReverse(double[,] dblDistance, int[] Sequence,
out double calDistance)
{
    int random1, random2;
    int[] Sequence2 = new int[Sequence.Length];
    Sequence.CopyTo(Sequence2, 0);
    Random rnd = new Random();
    do
    {
        random1 = rnd.Next(2, Sequence.Length + 1) - 1;
        random2 = rnd.Next(2, Sequence.Length + 1) - 1;
    } while (random1 == random2);
    int j = random2;
    if (random1 > random2)
    {
        for (int i = random1; i >= random2; i--)
        {
            Sequence2[i] = Sequence[j++];

```

```

    }
}
else
{
    for (int i = random1; i <= random2; i++)
    {
        Sequence2[i] = Sequence[j--];
    }
}
double dist1 = computeTotalDistance(dblDistance, Sequence);
double dist2 = computeTotalDistance(dblDistance, Sequence2);
if (dist1 <= dist2)
{
    calDistance = dist1;
    return Sequence;
}
else
{
    calDistance = dist2;
    return Sequence2;
}
}
private int[] newSequenceFromReverse(double[,] dblDistance, int[] Sequence,
int pos1, int pos2)
{
    int[] Sequence2 = new int[Sequence.Length];
    Sequence.CopyTo(Sequence2, 0);
    int j = pos2;
    if (pos1 > pos2)
    {
        for (int i = pos1; i >= pos2; i--)

```

```

        {
            Sequence2[i] = Sequence[j++];
        }
    }
    else
    {
        for (int i = pos1; i <= pos2; i++)
        {
            Sequence2[i] = Sequence[j--];
        }
    }

    if (computeTotalDistance(dblDistance, Sequence) <=
        computeTotalDistance(dblDistance, Sequence2))
    {
        return Sequence;
    }
    else
    {
        return Sequence2;
    }
}

private calResult TruckAndTransportFeeCal(int[] Sequence, Shop shop, Order
order, TruckDetails truck, int method = 1)
{
    calResult r = new calResult();
    bool[] CalFlag = new bool[Sequence.Length];
    CalFlag[0] = true;
    for (int i = 1; i < CalFlag.Length; i++) CalFlag[i] = false;

```



```

while (!isAllCal(CalFlag))
{
    double[] truckTotalVol;
    r.Truck6Cal.Add(TruckCal(method, Sequence, CalFlag,
order.Volume.ToArray(), truck.Truck6MaxVolume, truck.Truck6Num, out
truckTotalVol));

    r.Truck6TotalVol.Add(truckTotalVol);
    r.Truck4Cal.Add(TruckCal(method, Sequence, CalFlag,
order.Volume.ToArray(), truck.Truck4MaxVolume, truck.Truck4Num, out
truckTotalVol));
    r.Truck4TotalVol.Add(truckTotalVol);
}

r.TransportFee = 0;
double tempFee = 0;
// คำนวณค่าใช้จ่ายแบบ 4ล้อ
foreach (var t in r.Truck4Cal)
{
    r.Truck4FeeCal.Add(TransportFeeCal(t, shop.Distance,
truck.Truck4Fee.ToArray(), out tempFee, 2));
    r.TransportFee += tempFee;
}
foreach (var t in r.Truck6Cal)
{
    r.Truck6FeeCal.Add(TransportFeeCal(t, shop.Distance,
truck.Truck6Fee.ToArray(), out tempFee, 2));
    r.TransportFee += tempFee;
}

return r;
}

```

```
private double TruckAndTransportFeeCal2(int[] Sequence, Shop shop, Order order,
TruckDetails truck, int method = 1)
```

```
private List<int[]> TruckCal(int method, int[] Sequence, bool[] CalFlag, double[] Order,
double MaxVolume, int numTruck, out double[] TotalVolume)
```

```
private List<double> TransportFeeCal(List<int[]> TruckCal, double[,] dblDistance,
double[] TransportFee, out double TotalTransportFee, int method = 1)
```

```
private bool isAllCal(bool[] CalFlag)
private string getConfig(string key)
private void putConfig(string key, string value)
private string browseXlsFile()
private string browseCsvFile()
private static bool SequenceEqual(int[] seq1, int[] seq2)
private static string XlsConnection(string path)
private static string XlsxConnection(string path)
private void EnableAllButton(bool btnState = true)
private void btnSwapTable_Click(object sender, EventArgs e)
private void btnExit_Click(object sender, EventArgs e)
private void btnRandom_Click(object sender, EventArgs e)
private void radioRandom_Click(object sender, EventArgs e)
private void radioFix_Click(object sender, EventArgs e)
private void btnLoadFile_Click(object sender, EventArgs e)
    }
}
```