การประยุกต์ใช้วิธีการเชิงพันธุกรรมสำหรับปัญหาเส้นทางการขนส่งของโรงกำจัดซากไก่

นายนพรุจ สังข์แป้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2556
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย
The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

APPLICATION OF GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEM OF CHICKEN RENDERING PLANTS

Mr. Nopparuj Sangpan

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Industrial Engineering
Department of Industrial Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2013

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การประยุกต์ใช้วิธีการเชิงพันธุกรรมสำหรับปัญหาเส้นทางการขนส่งของโรงกำจัดซากไก่
โดย	นายนพรุจ สังข์แป้น
สาขาวิชา	วิศวกรรมอุตสาหการ
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รองศาสตราจารย์ สุทัศน์ รัตนเกื้อกังวาน

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร.บัณฑิต เอื้ออาภรณ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร.ปารเมศ ชุติมา)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ สุทัศน์ รัตนเกื้อกังวาน)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.อังศุมาลิน เสนจันทร์ฒิไชย)

.....กรรมการภายนอกมหาวิทยาลัย

(รองศาสตราจารย์ ดร.วันชัย ริจิรวนิช)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

นพรุจ สังข์แป้น : การประยุกต์ใช้วิธีการเชิงพันธุกรรมสำหรับปัญหาเส้นทางการขนส่งของโรงกำจัดซากไก่. (APPLICATION OF GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEM OF CHICKEN RENDERING PLANTS) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. สุทัศน์ รัตนเกื้อกังวาน, 164 หน้า.

การส่งออกเนื้อไก่สดมีการถูกกีดกันโดยข้ออ้างเรื่องการจัดการซากไก่ที่ไม่ถูกสุขลักษณะ ทำให้ทางภาครัฐมีดำริที่จะสร้างโรงกำจัดซากไก่ขึ้น 1 แห่ง โดยมีตัวอย่างของตำแหน่งที่ตั้งอยู่ 2 แห่ง คือ 1) บริเวณตำบลลำพญากลาง จังหวัดลพบุรี และ 2) ตำบลทับกวาง จังหวัดสระบุรี เพื่อรองรับปริมาณซากไก่และเศษซากไก่จากพื้นที่โดยรอบ สำหรับงานวิจัยนี้จะพิจารณาเส้นทางการขนส่งซากไก่และเศษซากไก่ จากศูนย์รวมซากไก่ต่างๆทั้ง 22 จุด โดยโรงกำจัดซากไก่มีปริมาณกำลังการผลิตของการผลิตทั้ง 2 สาย เท่ากับ 60,000 ตันต่อปี ซึ่งเมื่อคำนวณปริมาณทั้งหมดแล้วพบว่าจะมีปริมาณเกินกำลังการผลิตของสายการผลิตทั้ง 2 สาย จึงปรับปริมาณวัตถุดิบที่จะไปรับทั้ง 22 จุด โดยการปรับตามอัตราส่วน (proportional) เพื่อให้สามารถรองรับกำลังการผลิตได้ และงานวิจัยนี้มีวัตถุประสงค์เพื่อให้มีค่าใช้จ่ายรวมในการขนส่งให้ต่ำที่สุด ในการขนส่งจะเป็นการจ้างเหมารถบรรทุกซึ่งคิดค่าขนส่งเป็นอัตราเหมาจ่ายแบบขั้นบันไดตามระยะทาง (Stepwise Linear) โดยลักษณะปัญหานี้เป็นรูปแบบพิเศษของปัญหาการจัดเส้นทางเดินรถที่มีเงื่อนไขเวลาและความจุของรถ (Capacitated Vehicle Routing Problem with Time Window)

เนื่องจากปัญหาดังกล่าวมีความซับซ้อนแบบ NP-Hard จึงได้เสนอวิธีการหาคำตอบด้วยวิธีทางฮิวริสติก โดยแบ่งขั้นตอนเป็น 2 ช่วง ช่วงแรก คือ การสร้างคำตอบตั้งต้นด้วย Saving Algorithm และพัฒนาเพื่อหาคำตอบที่ดีขึ้นด้วย Genetic Algorithm ผลลัพธ์ที่ได้นำไปเปรียบเทียบกับ Saving Algorithm

จากการศึกษาพบว่า ตัวอย่างกรณีของโรงกำจัดซากไก่ตั้งที่ ตำบลลำพญากลาง สามารถลดค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมได้ 960,000 บาทต่อปี หรือ ร้อยละ 1.74 และในตัวอย่างกรณีโรงกำจัดซากไก่ตั้งที่ ตำบลทับกวาง สามารถลดค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมได้ 1,860,000 บาทต่อปี หรือ ร้อยละ 3.39 โดยผลการวิจัยนี้สามารถแก้ปัญหาที่มีความซับซ้อนได้อย่างรวดเร็ว และได้คำตอบที่ค่อนข้างดีในการหาคำตอบ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาควิชา วิศวกรรมอุตสาหการ ลายมือชื่อนิสิต .....

สาขาวิชา วิศวกรรมอุตสาหการ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก .....

ปีการศึกษา 2556

5470420121 : MAJOR INDUSTRIAL ENGINEERING
KEYWORDS: GENETIC ALGORITHM / VEHICLE ROUTING PROBLEM / CHICKEN RENDERING PLANTS

NOPPARUJ SANGPAN: APPLICATION OF GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEM OF CHICKEN RENDERING PLANTS. ADVISOR: ASSOC. PROF. SUTHAS RATANAKUAKANGWAN, 164 pp.

One of the conditions for non-tariff barriers for chicken export is the hygienic disposal of chicken carcasses. This has led to the planned construction of a public chicken rendering plant example in Lamphayaklang, Lopburi and in Tubkwang, Saraburi to accommodate chicken carcasses from 22 nodes nearby. With production capacity of 60,000 tons per year, which is a too big capacity, causes proportional adjustment. This research is intended to lowest the total expenses in transport. This is a Capacitated Vehicle Routing Problem with Time Window, which is one of the classical combinatorial problems. Furthermore, the stepwise linear objective function makes this problem even more challenging.

With HP-Hard complexity, the authors have proposed a method to tackle this problem in 2 phases : initialization with a saving algorithm and improvement with a genetic algorithm. The result compared to the Saving Algorithm.

The studies found that a public chicken rendering plant in Lamphayaklang, Lopburi can reduce 960,000 baht per year or 1.74 % and in Tubkwang, Saraburi 1,860,000 baht per year or 3.39 % in shipping expenses. This approach will not only deliver a good solution for this complex problem quickly, but it also can easily and rapidly solve for another new plant location.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Department: Industrial Engineering Student's Signature .....

Field of Study: Industrial Engineering Advisor's Signature .....

Academic Year: 2013

กิตติกรรมประกาศ

ในการทำวิทยานิพนธ์ฉบับนี้ ผู้เขียนขอกราบขอบพระคุณ รศ. สุทัศน์ รัตนเกื้อกังวาน อาจารย์ที่ปรึกษาในการทำวิทยานิพนธ์ ผศ. สุรพงษ์ ศิริกุลวัฒนา และ อ.น.สพ.ดร.ธราดล เหลืองทองคำ ซึ่งเป็นผู้ให้คำปรึกษา แนะนำแนวทางในการทำวิทยานิพนธ์จนสำเร็จลุล่วง และขอกราบขอบพระคุณประธานกรรมการสอบวิทยานิพนธ์ รศ.ดร. ปารเมศ ชุติมา และคณะกรรมการสอบวิทยานิพนธ์ ผศ.ดร.อังศุมาลิน เสนจันทร์ฒิไชย รศ.ดร.วันชัย ริจิรวนิช ที่ให้ความรู้ คำแนะนำ รวมถึงการแก้ไขข้อบกพร่องต่างๆ ของเนื้อหาการทำวิทยานิพนธ์จนเสร็จสมบูรณ์

ขอขอบคุณ คุณนพปฎล สกุลสม คุณอธิป น้อมศิริ และพี่ๆ เพื่อนๆ ร่วมรุ่นทุกท่านรวมไปถึงพี่ๆ เพื่อนๆ น้องๆ ในหน่วยปฏิบัติการวิจัยการบริหารอุตสาหกรรมและเทคโนโลยี ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ทุกท่านที่ช่วยเหลือให้คำแนะนำตั้งแต่เทคนิคขั้นตอนในการวิจัยจนถึงในการทำเล่มวิทยานิพนธ์ การช่วยตรวจทานการเขียนวิทยานิพนธ์ในเชิงภาษาและความเกี่ยวเนื่องของเนื้อหา รวมไปถึงความช่วยเหลือในด้านการจัดเตรียมเอกสารที่เกี่ยวข้อง และขอขอบคุณ สมาคมผู้ผลิตไก่เพื่อส่งออกไทย คณะผู้วิจัยโครงการจัดตั้งโรงงานกำจัดซากไก่ตายเพื่อการควบคุมโรค และกรมปศุสัตว์ สำหรับข้อมูลที่เกี่ยวข้องในงานวิจัย จนวิทยานิพนธ์เล่มนี้สำเร็จลุล่วง

สุดท้ายนี้ ใคร่ขอกราบขอบพระคุณบิดา มารดา และญาติผู้ใหญ่ที่เคารพทุกท่าน ซึ่งให้ความสนับสนุนในการศึกษาและให้กำลังใจแก่ผู้วิจัยเสมอมา จนสำเร็จการศึกษา

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฌ

สารบัญรูป ..... ฏ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของงานวิจัย 5

1.3 ขอบเขตของงานวิจัย 5

1.4 ขั้นตอนการดำเนินงาน 5

1.5 ประโยชน์ที่คาดว่าจะได้รับ 5

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 6

2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง 6

2.1.1 ปัญหาการจัดเส้นทางการเดินรถ (Vehicle Routing Problem ; VRP) 6

2.1.2 อัลกอริทึมแบบประหยัด (Saving Algorithm) 9

2.1.3 วิธีการเชิงพันธุกรรม (Genetic Algorithm) 10

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง 23

บทที่ 3 วิธีการดำเนินงานวิจัย 25

3.1 การเตรียมข้อมูลในงานวิจัย 25

3.2 แบบจำลองทางคณิตศาสตร์ (Mathematical model) 33

3.3 การสร้างคำตอบตั้งต้น 39

บทที่ 4 การปรับปรุงเส้นทางโดยวิธีการเชิงพันธุกรรม 51

4.1 ขั้นตอนของ Genetic Algorithm 51

4.2 ผลการวิจัย 61

หน้า

บทที่ 5 การสรุปผลการวิจัย และ ข้อเสนอแนะ....64
5.1 สรุปผลการวิจัย....64
5.2 ข้อเสนอแนะ....64
รายการอ้างอิง....65
ภาคผนวก....67
ภาคผนวก ก ตัวอย่าง การปรับปรุงเส้นทางตั้งต้นโดย Genetic Algorithm....68
ภาคผนวก ข รหัสคำสั่งที่ใช้ในโปรแกรม IBM ILOG CPLEX OPTIMIZATION STUDIO สำหรับการสร้างแบบจำลองทางคณิตศาสตร์....84
ภาคผนวก ค รหัสคำสั่งของ Saving Algorithm และ Genetic Algorithm ที่ใช้ใน JAVA Script....89
ประวัติผู้เขียนวิทยานิพนธ์....164

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญตาราง

หน้า

ตารางที่ 1-1 พิกัดที่ตั้งของโรงกำจัดซากไก่ และศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด.....2

ตารางที่ 1-2 ปริมาณซากไก่และเศษซากไก่ (โครงไก่) ในแต่ละจุดที่จะส่งเป็นวัตถุดิบให้โรงกำจัดซากไก่.....4

ตารางที่ 2-1 ความหมายของคำศัพท์ทางพันธุกรรม.....11

ตารางที่ 3-1 ระยะทางระหว่างแต่ละจุด กรณีที่ตั้งโรงกำจัดซากไก่ที่ ต.ลำพญากลาง จ.ลพบุรี.....27

ตารางที่ 3-2 ระยะทางระหว่างแต่ละจุด กรณีที่ตั้งโรงกำจัดซากไก่ที่ ต.ทับกวาง จ.สระบุรี.....28

ตารางที่ 3-3 ปริมาณซากไก่และเศษซากไก่ (โครงไก่) ในแต่ละจุดที่จะส่งเป็นวัตถุดิบให้โรงกำจัดซากไก่ที่ปรับปริมาณวัตถุดิบที่จะไปรับแต่ละจุดตามอัตราส่วน.....30

ตารางที่ 3-4 อัตราค่าขนส่งของรถบรรทุก 10 ล้อ แบบห้องเย็นขนาด 12 ตัน.....32

ตารางที่ 3-5 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 5 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง).....36

ตารางที่ 3-6 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 4 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง).....36

ตารางที่ 3-7 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 3 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง).....37

ตารางที่ 3-8 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 5 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง).....37

ตารางที่ 3-9 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 4 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง).....38

ตารางที่ 3-10 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 3 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง).....38

ตารางที่ 3-11 คำนวณค่าความประหยัดระหว่างแต่ละจุด กรณีที่ตั้งโรงกำจัดซากไก่ที่ ต.ลำพญากลาง จ.ลพบุรี.....40

ตารางที่ 3-12 คำนวณค่าความประหยัดระหว่างแต่ละจุด กรณีที่ตั้งโรงกำจัดซากไก่ที่ ต.ทับกวาง จ.สระบุรี.....41

ตารางที่ 3-13 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 5 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง).....44

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

หน้า

ตารางที่ 3-14 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 4 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง).....44

ตารางที่ 3-15 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 3 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง).....45

ตารางที่ 3-16 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 5 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง).....45

ตารางที่ 3-17 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 4 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง).....46

ตารางที่ 3-18 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 3 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง).....46

ตารางที่ 3-19 เปรียบเทียบผลลัพธ์ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมดและระยะทางทั้งหมดของการไปรับซากและเศษซากไก่ ระหว่างวิธีการหาค่าที่ดีที่สุดกับ Saving Algorithm (กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง).....47

ตารางที่ 3-20 เปรียบเทียบผลลัพธ์ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมดและระยะทางทั้งหมดของการไปรับซากและเศษซากไก่ ระหว่างวิธีการหาค่าที่ดีที่สุดกับ Saving Algorithm (กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง).....47

ตารางที่ 3-21 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง).....48

ตารางที่ 3-22 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง).....49

ตารางที่ 4-1 ตัวอย่างการสร้าง Roulette Wheel.....52

ตารางที่ 4-2 ตัวอย่างการคัดเลือกโครโมโซมด้วยวิธีการ Binary Tournament Selection.....53

ตารางที่ 4-3 แสดงค่าพารามิเตอร์ของ Genetic Algorithm สำหรับการจัดเส้นทางของโรงกำจัดซากไก่.....53

ตารางที่ 4-4 ผลการทดลองความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm) (สำหรับการเดินทางไปรับซากและเศษซากไก่ กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง).....56

ตารางที่ 4-5 ผลการทดลองความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm) (สำหรับการเดินทางไปรับซากและเศษซากไก่ กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง).....58

ตารางที่ 4-6 ผลลัพธ์จาก Genetic Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง).....61

หน้า

ตารางที่ 4-7 ผลลัพธ์จาก Genetic Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง).....62

ตารางที่ 4-8 เปรียบเทียบผลลัพธ์ค่าใช้จ่ายแบบเหมาจ่ายรวมและระยะทางรวม ของกรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง และ ต.ทับกวาง.....63

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญรูป

หน้า

รูปที่ 1-1 ตำแหน่งของโรงกำจัดซากไก่และศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีตัวอย่างโรงกำจัดซากตั้งที่ ต.ลำพญากลาง จ.ลพบุรี).....3
รูปที่ 1-2 ตำแหน่งของโรงกำจัดซากไก่และศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีตัวอย่างโรงกำจัดซากตั้งที่ ต.ทับกวาง จ.สระบุรี).....3
รูปที่ 2-1 การจัดเส้นทางเดินรถที่มีคลังสินค้า 1 แห่ง.....6
รูปที่ 2-2 ผลการจัดเส้นทางที่มีการเกิดเส้นทางย่อย (Sub-tour).....8
รูปที่ 2-3 การเปรียบเทียบการเดินทางแบบสองรอบและหนึ่งรอบ.....9
รูปที่ 2-4 ลำดับขั้นตอนวิธีการเชิงพันธุกรรม.....12
รูปที่ 2-5 ค่าโครโมโซมของค่าความต้องการสินค้า.....12
รูปที่ 2-6 การออกแบบโครโมโซมแบบลำดับ.....13
รูปที่ 2-7 การออกแบบโครโมโซมแบบใช้ค่าหรือเครื่องหมายจริง.....13
รูปที่ 2-8 การสุ่มเลือกประชากรตั้งต้น.....14
รูปที่ 2-9 การกำหนดจุดตัดการแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบหนึ่งจุด.....15
รูปที่ 2-10 การถ่ายทอดยีนส์ที่ถูกตัดในช่วงแรกมายังโครโมโซมลูก.....15
รูปที่ 2-11 การแลกเปลี่ยนยีนข้ามโครโมโซมแบบหนึ่งจุด โดยทำการตัดตัวที่ซ้ำออก.....15
รูปที่ 2-12 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบหนึ่งจุด โดยทำการถ่ายทอดยีนส์ที่ถูกตัดตัวซ้ำแล้ว.....15
รูปที่ 2-13 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบหนึ่งจุด.....15
รูปที่ 2-14 การกำหนดจุดตัดการแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด.....16
รูปที่ 2-15 การถ่ายทอดยีนส์ที่ถูกตัดในช่วงมายังโครโมโซมลูก.....16
รูปที่ 2-16 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด โดยทำการตัดตัวที่ซ้ำออก.....16
รูปที่ 2-17 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด โดยทำการถ่ายทอดยีนส์ที่ถูกตัดตัวซ้ำแล้ว.....16
รูปที่ 2-18 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด.....16
รูปที่ 2-19 การกำหนดจุดตัดการแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX.....17
รูปที่ 2-20 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX โดยทำการสลับยีนส์ข้ามโครโมโซมของช่วงที่ถูกตัด.....17

หน้า

รูปที่ 2-21 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX โดยทำการสร้างแผนภาพความสัมพันธ์.....17
รูปที่ 2-22 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX โดยทำการเปลี่ยนค่ายีนส์จากแผนภาพความสัมพันธ์.....17
รูปที่ 2-23 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX.....18
รูปที่ 2-24 การกำหนดจุดตัดการแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ WMX.....18
รูปที่ 2-25 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ WMX โดยทำการสร้างแผนภาพความสัมพันธ์.....18
รูปที่ 2-26 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ WMX โดยทำการเปลี่ยนค่ายีนส์จากแผนภาพความสัมพันธ์.....18
รูปที่ 2-27 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ WMX.....19
รูปที่ 2-28 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก ขั้นตอนการเลือกโครโมโซม.....19
รูปที่ 2-29 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก ขั้นตอนการสุ่มตำแหน่งยีนส์ที่จะทำการแทรก.....19
รูปที่ 2-30 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก ขั้นตอนการสุ่มเลือกยีนส์ที่จะนำเข้ามาแทรก.....19
รูปที่ 2-31 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก (Insertion Mutation).....20
รูปที่ 2-32 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง ขั้นตอนการสุ่มเลือกโครโมโซม.....20
รูปที่ 2-33 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง ขั้นตอนการสุ่มตำแหน่งยีนส์ที่จะทำการเคลื่อนตำแหน่ง.....20
รูปที่ 2-34 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง ขั้นตอนการสุ่มเลือกช่วงยีนส์ที่จะ นำเข้ามาเคลื่อนตำแหน่ง.....20
รูปที่ 2-35 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง (Displacement Mutation).....20
รูปที่ 2-36 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง ขั้นตอนการสุ่มเลือกโครโมโซม.....21
รูปที่ 2-37 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง ขั้นตอนการสุ่มตำแหน่งยีนส์.....21

หน้า

รูปที่ 2-38 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง
ขั้นตอนการสลับตำแหน่งยีนส์..... 21

รูปที่ 2-39 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง.....21

รูปที่ 3-1 แผนผังขั้นตอนการหาคำตอบตั้งต้นโดย Saving Algorithm.....43

รูปที่ 4-1 ตัวอย่างการสร้าง Roulette Wheel.....52

รูปที่ 4-2 แสดงผลของราคาค่าขนส่งแบบเหมาจ่ายรวมต่อจำนวนรอบการทดลอง
(สำหรับการเดินทางไปรับซากและเศษซากไก่ กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง).....54

รูปที่ 4-3 แสดงผลของราคาค่าขนส่งแบบเหมาจ่ายรวมต่อจำนวนรอบการทดลอง
(สำหรับการเดินทางไปรับซากและเศษซากไก่ กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง).....55

รูปที่ 4-4 แสดงการวิเคราะห์ค่าความแปรปรวนแบบทางเดียว (One way Anova)
ระหว่างค่าใช้จ่ายรวมกับข้อมูลแต่ละครั้งและระหว่างระยะทางรวมกับข้อมูลแต่ละครั้ง
(สำหรับการเดินทางไปรับซากและเศษซากไก่ กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง).....57

รูปที่ 4-5 แสดงการวิเคราะห์ค่าความแปรปรวนแบบทางเดียว (One way Anova)
ระหว่างค่าใช้จ่ายรวมกับข้อมูลแต่ละครั้งและระหว่างระยะทางรวมกับข้อมูลแต่ละครั้ง
(สำหรับการเดินทางไปรับซากและเศษซากไก่ กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง).....59

รูปที่ 4-6 แผนผังแสดงขั้นตอนการปรับปรุงคำตอบตั้งต้นโดย Genetic Algorithm.....60

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1
บทนำ

ในบทนี้จะกล่าวถึงที่มาและความสำคัญของปัญหาของการจัดตั้งโรงกำจัดซากไก่ รวมไปถึงวัตถุประสงค์ของงานวิจัย ขอบเขตของงานวิจัย ประโยชน์ที่คาดว่าจะได้รับ และขั้นตอนการดำเนินงานของงานวิจัย

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันอุตสาหกรรมการเลี้ยงไก่เนื้อนับว่าเป็นอุตสาหกรรมการเกษตรสำคัญอย่างหนึ่งของประเทศไทย โดยมีการขยายตัวเพิ่มมากขึ้นเนื่องจากตลาดต่างประเทศเปิดตลาดให้ประเทศไทยสามารถส่งออกเนื้อไก่สดได้ ส่งผลทำให้มีปริมาณการเลี้ยงไก่เพิ่มมากขึ้นตามความต้องการของตลาด ในทางกลับกันการเพิ่มปริมาณการเลี้ยงก็ส่งผลต่อความเสี่ยงของการควบคุมโรคและการจัดการฟาร์ม โดยในปัจจุบันการจัดการซากไก่ที่ตายในฟาร์มยังคงใช้วิธีการฝังกลบหรือเผาซาก นอกจากนี้ยังมีการลักลอบนำซากไก่ในไปขายเพื่อบริโภค ส่งผลต่อเกิดโอกาสการแพร่กระจายของเชื้อโรคและสุขภาพร่างกายของประชาชนได้ ดังนั้นทางสมาคมผู้ส่งออกไก่ไทยจึงเกิดแนวความคิดที่จะส่งเสริมให้มีการจัดตั้งโรงกำจัดซากไก่ขึ้นเพื่อกำจัดซากไก่จากฟาร์มและโครงไก่จากโรงฆ่าให้เป็นไปอย่างถูกสุขลักษณะและไม่กระทบกับสิ่งแวดล้อม นอกจากนั้นยังได้กากเนื้อและกระดูกป่นซึ่งเป็นผลิตภัณฑ์ของโรงกำจัดซากไก่มาผลิตเป็นอาหารสัตว์เลี้ยง ทำให้สามารถลดการนำเข้าวัตถุดิบดังกล่าวจากต่างประเทศได้

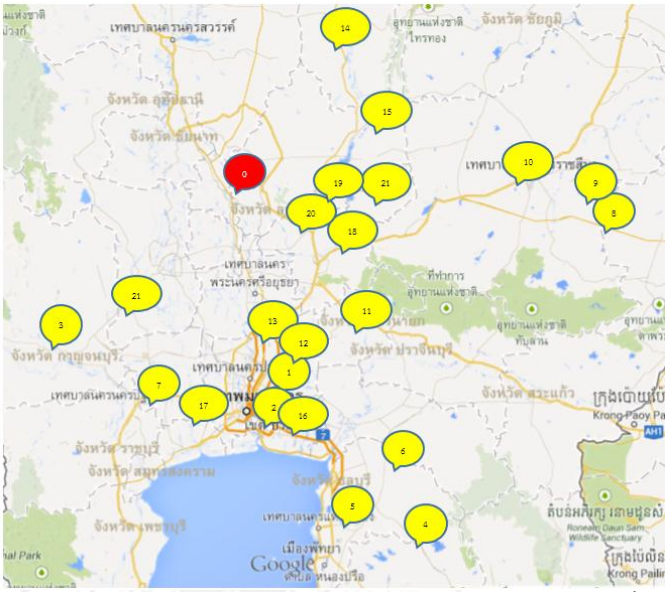
แต่อย่างไรก็ดี โดยลักษณะโรงกำจัดซากไก่จะมีขนาดใหญ่และก่อตั้งเพียงแห่งเดียวในประเทศ ดังนั้นปัญหาที่ตามมาส่วนหนึ่งของการจัดตั้งโรงกำจัดซากไก่ คือ ปัญหาการขนส่งของซากไก่ตายจากศูนย์รวมซากไก่จากฟาร์มและเศษซากไก่จากโรงฆ่าต่างๆทั้ง 22 จุด เพื่อให้มีความสม่ำเสมอและสามารถรองรับกำลังการผลิตของโรงกำจัดซากไก่ได้ โดยพิกัดที่ตั้งของโรงกำจัดซากไก่ และศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด จะแสดงจากพิกัดในระบบ GPS (หมายเลข EST. ย่อมาจาก Establishment Approved Number ในแต่ละจุดของลูกค้า แสดงถึง ลำดับหมายเลขโรงงานที่ได้รับการอนุมัติให้สามารถส่งออกต่างประเทศได้) โดยมีรายละเอียด ดังแสดงในตารางที่ 1-1 (ข้อมูลจากสำนักตรวจสอบฯ กรมปศุสัตว์)

ตารางที่ 1-1 พิกัดที่ตั้งของโรงกำจัดซากไก่ และศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด

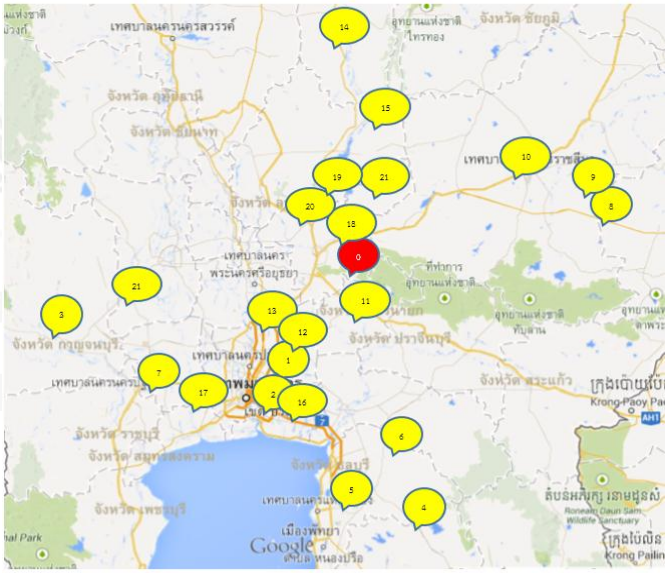
รหัสที่ใช้อ้างอิง	หน่วยงาน	EST	จังหวัด	พิกัดที่ตั้ง
0	ตัวอย่าง โรงกำจัดซากไก่ กรณี 1	-	ลพบุรี (ต.ลำพญากลาง)	101.382 E / 15.149 N
0	ตัวอย่าง โรงกำจัดซากไก่ กรณี 2	-	สระบุรี (ต.ทับกวาง)	101.042 E / 14.642 N
1	ซีพีเอฟ	3	กรุงเทพ	100.797 E / 13.812 N
2	ซีพีเอฟ	18	กรุงเทพ	100.373 E / 13.395 N
3	ไทยฟู้ดกรุ๊ป	119	กาญจนบุรี	99.801 E / 14.061 N
4	GFN	201	ชลบุรี	101.309 E / 13.159 N
5	ไทยโพลทรีกรุ๊ป	109	ชลบุรี	100.973 E / 13.128 N
6	พนัส	79	ชลบุรี	101.209 E / 13.398 N
7	แหลมทอง	6	นครปฐม	100.299 E / 13.727 N
8	ซีพีเอฟ	139	นครราชสีมา	102.214 E / 14.744 N
9	คาร์กิล มีท	165	นครราชสีมา	102.185 E / 14.681 N
10	แหลมทอง	70	นครราชสีมา	101.837 E / 14.889 N
11	โกลเด้น ฟู้ดส์	53	นครนายก	101.010 E / 14.054 N
12	โกลเด้น ฟู้ดส์	34	ปทุมธานี	100.742 E / 13.961 N
13	เซนทาโก	7	ปทุมธานี	100.373 E / 14.213 N
14	สหฟาร์ม	129	เพชรบูรณ์	101.070 E / 15.846 N
15	สหฟาร์ม	44	ลพบุรี	101.025 E / 15.097 N
16	GFPT	11	สมุทรปราการ	100.774 E / 13.591 N
17	เบทาโกร	10	สมุทรสาคร	100.306 E / 13.715 N
18	ซีพีเอฟ	23	สระบุรี	100.988 E / 14.575 N
19	เบทาโกร	49	สระบุรี	100.912 E / 14.829 N
20	คาร์กิล มีท	32	สระบุรี	100.840 E / 14.667 N
21	ซันฟู้ด	25	สระบุรี	101.116 E / 14.798 N
22	F&F	160	สุพรรณบุรี	99.869 E / 14.149 N

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

โดยพิกัดที่ตั้งของโรงกำจัดซากไก่ และศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด มาทำการแสดงตำแหน่ง โดยจุดสีแดงแสดงถึงตำแหน่งของโรงกำจัดซากไก่ และจุดสีเหลืองแสดงถึงตำแหน่งศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด ที่อยู่ในบริเวณรอบๆโรงกำจัดซากไก่ ดังแสดงในรูปที่ 1-1 และ 1-2



รูปที่ 1-1 ตำแหน่งของโรงกำจัดซากไก่และศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด
(กรณีตัวอย่างโรงกำจัดซากตั้งที่ ต.ลำพญากลาง จ.ลพบุรี)



รูปที่ 1-2 ตำแหน่งของโรงกำจัดซากไก่และศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด
(กรณีตัวอย่างโรงกำจัดซากตั้งที่ ต.ทับกวาง จ.สระบุรี)

จากรูปที่ 1-1 และ 1-2 พบว่า ตำแหน่งของศูนย์รวมซากและเศษซากไก่ทั้งหมด 22 จุด มีการกระจายตัวในตำแหน่งต่างๆที่ไม่สม่ำเสมอกัน และในบางจุดมีระยะทางจากตำแหน่งของโรงกำจัดซากไก่ที่มาก จึงเกิดความซับซ้อนในการจัดเส้นทางขนส่งที่มากขึ้น

โดยในส่วนของปริมาณซากและเศษซากไก่ในแต่จุดที่จะส่งเป็นวัตถุดิบให้โรงกำจัดซากไก่ พบว่า มีปริมาณที่แตกต่างกันในแต่ละจุดเนื่องจากปริมาณไก่ที่เข้าโรงเชือดในแต่ละศูนย์รวมมีค่าแตกต่างกัน ดังแสดงในตารางที่ 1-2 (ข้อมูลจากสมาคมผู้ผลิตไก่เพื่อส่งออกไทย)

ตารางที่ 1-2 ปริมาณซากไก่และเศษซากไก่ (โครงไก่) ในแต่ละจุดที่จะส่งเป็นวัตถุดิบให้โรงกำจัดซากไก่

จุดลูกค้า	EST	จังหวัด	ปริมาณไก่ที่เข้าโรงเชือด (ตัว/วัน)
1	3	กรุงเทพ	180,000
2	18	กรุงเทพ	60,000
3	119	กาญจนบุรี	340,000
4	201	ชลบุรี	100,000
5	109	ชลบุรี	110,000
6	79	ชลบุรี	130,000
7	6	นครปฐม	70,000
8	139	นครราชสีมา	260,000
9	165	นครราชสีมา	110,000
10	70	นครราชสีมา	130,000
11	53	นครนายก	90,000
12	34	ปทุมธานี	90,000
13	7	ปทุมธานี	90,000
14	129	เพชรบูรณ์	250,000
15	44	ลพบุรี	310,000
16	11	สมุทรปราการ	130,000
17	10	สมุทรสาคร	130,000
18	23	สระบุรี	240,000
19	49	สระบุรี	270,000
20	32	สระบุรี	220,000
21	25	สระบุรี	90,000
22	160	สุพรรณบุรี	50,000
รวม			**3,450,000**

ดังนั้นจากข้อมูลทั้งตำแหน่งศูนย์รวมซากและเศษซากทั้งหมด 22 จุด ที่มีการกระจายตัวที่ไม่สม่ำเสมอกัน และข้อมูลปริมาณซากไก่และเศษซากไก่ (โครงไก่) ในแต่ละจุดที่จะส่งเป็นวัตถุดิบให้โรงกำจัดซากไก่ที่มีค่าแตกต่างกันในแต่ละจุด จึงทำให้ความซับซ้อนในการจัดเส้นทางการขนส่งซากไก่ ดังนั้นการกำหนดเส้นทางเพื่อให้เกิดความคุ้มค่าสูงสุดทั้งในด้านค่าใช้จ่ายและระยะทางการขนส่งจึงมีความสำคัญ เพื่อประกอบการตัดสินใจในการลงทุนในด้านการขนส่งซากไก่และเศษซากไก่ได้อย่างถูกต้องในอนาคต

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.2 วัตถุประสงค์ของงานวิจัย

งานวิจัยนี้มีจุดประสงค์เพื่อสร้างเส้นทางการขนส่งซากไก่และเศษซากไก่สำหรับโรงกำจัดซากไก่ให้มีประสิทธิภาพทั้งในด้านค่าใช้จ่ายและระยะทาง

1.3 ขอบเขตของงานวิจัย

ทำการศึกษาเพื่อสร้างเส้นทางการขนส่งซากไก่และเศษซากไก่ โดยเริ่มต้นโรงกำจัดซากไก่ (Rendering Plants) ไปรับยังศูนย์รวมซากและเศษซากทั้งหมด 22 จุด และกลับไปสิ้นสุดที่โรงกำจัดซากไก่ โดยลำดับเส้นทางการขนส่งของรถเหมือนกันทุกวัน

1.4 ขั้นตอนการดำเนินงาน

วิธีการดำเนินงานวิจัยมีขั้นตอนดังนี้

1. ศึกษาทฤษฎี บทความ และงานวิจัยที่เกี่ยวข้องกับการสร้างเส้นทางการขนส่ง เพื่อนำมาประยุกต์กับงานวิจัย
2. ศึกษาข้อมูลพื้นฐานและทำการรวบรวมข้อมูลต่างๆที่เกี่ยวข้องในงานวิจัย
3. ทำการสร้างและประมวลผลแบบจำลองทางคณิตศาสตร์
4. หาเส้นทางคำตอบตั้งต้นโดยวิธี Saving Algorithm
5. ปรับปรุงคำตอบตั้งต้นโดยการประยุกต์ใช้ Genetic Algorithm โดยมีการทดลองออกแบบจำนวนรอบการทดสอบ และการทดลองของพารามิเตอร์ความน่าจะเป็นของการครอสโอเวอร์ (Crossover) และมิวเตชั่น (Mutation)
6. เปรียบเทียบและวิเคราะห์ผลลัพธ์ที่ได้
7. สรุปผลการศึกษา และจัดทำรูปเล่มวิทยานิพนธ์

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับมีดังต่อไปนี้

1. สามารถหาเส้นทางเดินรถและจำนวนรถขนส่งที่ต้องใช้เพื่อลดความสูญเสียเวลาและค่าใช้จ่ายในการขนส่งซากและเศษซากไก่เข้าสู่โรงกำจัดซากไก่
2. เพื่อเป็นข้อมูลประกอบในการตัดสินใจการลงทุนของผู้ประกอบการในการขนส่งซากและเศษซากไก่ที่เข้าสู่โรงกำจัดซากไก่
3. เพื่อเพิ่มความเชื่อมั่นของต่างชาติด้านโรคระบาดในการส่งออกไก่สดออกนอกประเทศ อันเนื่องจากจากการมีระบบการจัดการการกำจัดซากไก่และการขนส่งซากไก่ที่ดี
4. เพื่อเป็นแนวทางในการสร้างเส้นทางการขนส่งสำหรับงานวิจัยอื่นๆต่อไป

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงแนวคิดและทฤษฎีที่เกี่ยวข้อง รวมไปถึงเอกสารและงานวิจัยที่เกี่ยวข้อง เพื่อที่จะนำไปสู่ขั้นตอนวิธีการดำเนินงานวิจัยต่อไป

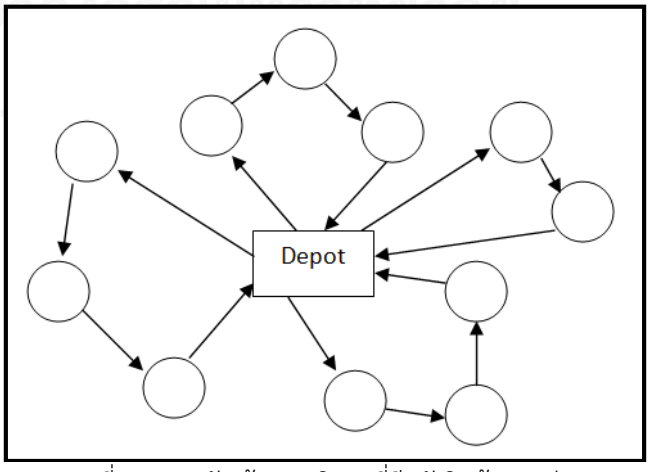
2.1 แนวคิดและทฤษฎีที่เกี่ยวข้อง

แนวคิดและทฤษฎีที่เกี่ยวข้องประกอบด้วย

1. ปัญหาการจัดเส้นทางการเดินรถ (Vehicle Routing Problem : VRP)
2. อัลกอริทึมแบบประหยัด (Saving Algorithm)
3. วิธีการเชิงพันธุกรรม (Genetic Algorithm)

2.1.1 ปัญหาการจัดเส้นทางการเดินรถ (Vehicle Routing Problem : VRP) [1]

เป็นการออกแบบเส้นทางเดินรถในแต่ละรอบการวิ่งให้มีความเหมาะสมที่สุดทั้งในเรื่องของค่าใช้จ่ายและข้อจำกัดต่างๆที่เกี่ยวข้องด้วย เช่น ปริมาณความจุของรถบรรทุก เวลาในการเดินรถ เวลาในการขนส่งสินค้าขึ้นหรือลงจากรถ โดยเส้นทางการขนส่งในแต่ละรอบการวิ่งจะเริ่มต้นจากคลังสินค้า (depot) แล้วเดินทางไปยังกลุ่มลูกค้า โดยที่ทราบจำนวนและตำแหน่งต่างๆของลูกค้าในแต่ละจุด และทราบปริมาณของสินค้าล่วงหน้าที่แน่นอน แล้วเดินทางกลับมาสิ้นสุดที่คลังสินค้า (depot) โดยมีข้อจำกัดว่าปริมาณสินค้าที่ต้องไปรับหรือไปส่งต้องไม่เกินความสามารถในการบรรทุกสูงสุดของรถขนส่งในรอบการวิ่งนั้นๆและลูกค้าแต่ละรายจะได้รับหรือจ่ายสินค้าจากการขนส่งจากรอบการวิ่งเดียวหรือหลายรอบก็ได้ รวมไปถึงเวลาในการเดินรถในแต่ละรอบการวิ่งนั้นมีจำกัดด้วย



รูปที่ 2-1 การจัดเส้นทางเดินรถที่มีคลังสินค้า 1 แห่ง

วัตถุประสงค์หลักของการออกแบบเส้นทางเดินรถเพื่อให้เกิดความเหมาะสมสูงสุด มี 4 ประการ [2] ดังนี้

1. เพื่อลดค่าใช้จ่ายต้นทุนคงที่ (Fixed Cost) เช่น สามารถลดจำนวนรถขนส่งสินค้าหรือจำนวนรอบในการขนส่งแต่ละครั้ง และรวมไปถึงการลดการจ้างพนักงานในการขับรถในแต่ละครั้งไปด้วย เป็นต้น
2. เพื่อลดค่าใช้จ่ายต้นทุนแปรผัน (Variable Cost) เช่น สามารถลดระยะทางและเวลาในการขนส่งในแต่ละรอบได้ และรวมไปถึงการลดต้นทุนแปนผันที่น้อยลงไปด้วย เช่น ค่าเชื้อเพลิง และค่าบำรุงรักษา เป็นต้น
3. เพื่อลดค่าใช้จ่ายต้นทุนทั้งหมด (Total Cost) ให้น้อยที่สุด ได้แก่ ค่าใช้จ่ายที่เกิดจากต้นทุนคงที่และค่าใช้จ่ายที่เกิดจากต้นทุนแปรผัน
4. การออกแบบเส้นทางการขนส่งที่แน่นอนเพื่อเพิ่มความพึงพอใจและความเชื่อมั่นให้กับลูกค้าที่รับบริการ

โดยรูปแบบพื้นฐานของปัญหาการจัดเส้นทางเดินรถที่มีคลังสินค้าเพียง 1 แห่ง สามารถอธิบายด้วยแบบจำลองทางคณิตศาสตร์ [3] ได้ดังนี้

$$Minimize \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{v=1}^{NV} C_{ij}x_{ij}^{v} \tag{1}$$

$$\sum_{v=1}^{NV}\sum_{i=o}^{n} x_{ij}^{v} \geq 1 \qquad ;(j=0,1,...,n) \tag{2}$$

$$\sum_{i=0}^{n} x_{ip}^{v} - \sum_{j=0}^{n} x_{pj}^{v} = 0 \qquad ;(p=0,1,...,n)\ (v=1,2,...,NV) \tag{3}$$

$$\sum_{v=1}^{NV} y_{iv} = 1 \qquad ;(i=1,2,...,n) \tag{4}$$

$$\sum_{i=1}^{n} d_i y_{iv} \leq Q_v \qquad ;(v=1,2,...,NV) \tag{5}$$

$$y_{iv} \leq \sum_{j=0}^{n} x_{ji}^{v} \qquad ;(i=1,2,...,n)\ (v=1,2,...,NV) \tag{6}$$

$$X \in S \tag{7}$$

$$x_{ij}^{v} = 0,1 \qquad ;(i=0,1,...,n)(j=0,1,...,n)(v=1,2,...,NV) \tag{8}$$

$$y_{iv} \geq 0 \qquad ;(i=1,2,...,n)(v=1,2,...,NV) \tag{9}$$

เมื่อ C_{ij} = ค่าใช้จ่ายในการขนส่งระหว่างการเดินทางจากจุด i ไปยังจุด j

d_i = จำนวนสินค้าของลูกค้าที่จุด i

Q_v = ความสามารถในการบรรทุกของรถรอบการวิ่งที่ v

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$X_{ij}^{v} = \begin{cases} 1 & \text{เมื่อมีการจัดรถรอบการวิ่งที่ } v \text{ ให้ไปรับสินค้าจาก } i \text{ ไปยังจุด } j \\ 0 & \text{เมื่อเป็นกรณีอื่นๆ} \end{cases}$$

y_{iv} = สัดส่วนจำนวนสินค้าของลูกค้าที่จุด i ที่ถูกขนส่งโดยรถรอบการวิ่งที่ v

NV = จำนวนรอบการวิ่งทั้งหมด

S = เซตของรอบทั้งหมดที่อยู่ใน N ชุด ซึ่งรวมถึงคลังสินค้า (depot)
(จุด 0 แทนคลังสินค้า)

จากแบบจำลองทางคณิตศาสตร์สรุปความได้ดังนี้

ความสัมพันธ์ที่ 1 แสดงถึง วัตถุประสงค์เพื่อให้ได้ค่าใช้จ่ายในการขนส่งที่ต่ำที่สุด

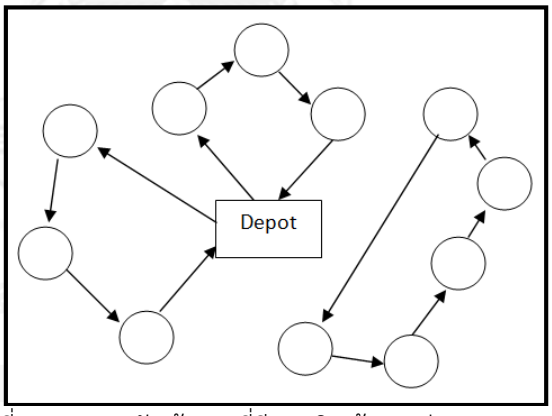
ความสัมพันธ์ที่ 2 แสดงถึง ปริมาณสินค้าในแต่ละจุดของลูกค้าจะถูกรับอย่างน้อย 1 รอบการวิ่ง

ความสัมพันธ์ที่ 3 แสดงถึง การบังคับให้รถวิ่งออกจากคลังสินค้าออกไปรับสินค้าแล้ววิ่งกลับมายังคลังสินค้าเสมอ

ความสัมพันธ์ที่ 4 แสดงถึง ปริมาณสินค้าในแต่ละจุดของลูกค้าจะถูกรับทั้งหมด

ความสัมพันธ์ที่ 5 แสดงถึง การขนส่งในแต่ละรอบการวิ่งจะไม่สามารถบรรทุกเกินความสามารถของรถได้

ความสัมพันธ์ที่ 6 และ 7 แสดงถึง การป้องกันการเดินรถย่อย (Sub-tour) โดยรถในแต่ละรอบการวิ่งนั้นๆจะมีการเดินทางออกจากคลังสินค้าและกลับไปยังสินค้าเสมอ



รูปที่ 2-2 ผลการจัดเส้นทางที่มีการเกิดเส้นทางย่อย (Sub-tour)

ความสัมพันธ์ที่ 8 แสดงถึง เงื่อนไขของการเดินทางไปรับสินค้าโดยแสดงเป็นแบบ binary

ความสัมพันธ์ที่ 9 แสดงถึง จำนวนสัดส่วนของสินค้าของลูกค้าต้องมากกว่าหรือเท่ากับ 0

ในการจัดประเภทรูปแบบของการจัดเส้นทางสามารถแบ่งออกได้ [4] ดังต่อไปนี้

1. ปัญหาการจัดเส้นทางเดินรถแบบมีขีดจำกัดของความสามารถในการบรรทุกของรถ (Capacitated Vehicle Routing Problem)
2. ปัญหาการจัดเส้นทางเดินรถแบบมีขอบข่ายของเวลา (Vehicle Routing Problem with Fixed Time)

3. ปัญหาการจัดเส้นทางเดินรถแบบมีขอบข่ายของเวลาในการรับส่งสินค้า (Vehicle Routing Problem with Time Window)

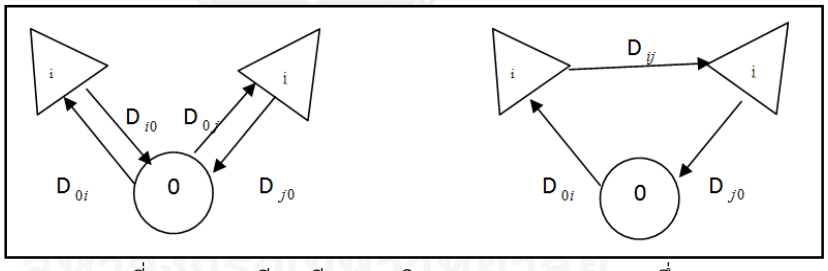
4. ปัญหาการจัดเส้นทางเดินรถแบบมีขนาดรถที่หลากหลายและมีความจุที่แตกต่างกัน (Heterogeneous Fleet Vehicle Routing Problem)

5. ปัญหาการจัดเส้นทางเดินรถแบบมีความหลากหลายในการบรรทุกภายในรถหนึ่งคัน (Multi-compartment Vehicle Routing Problem)

6. ปัญหาการจัดเส้นทางเดินรถแบบมีหลายคลังสินค้า (Multi depot Vehicle Routing Problem)

2.1.2 อัลกอริทึมแบบประหยัด (Saving Algorithm)

Clarke and Wright [5] ได้เสนอวิธีการแก้ปัญหาการจัดเส้นทาง เริ่มต้นโดยการให้รถแต่ละรอบการวิ่งออกจากคลังสินค้าเดินทางไปรับสินค้าจากลูกค้าเพียงจุดเดียวแล้วเดินทางกลับมาสิ้นสุดที่คลังสินค้า ซึ่งใช้รอบการวิ่งรถที่มาก จึงได้พัฒนาให้การเดินรถในแต่ละรอบการวิ่งให้ไปรับสินค้าจากลูกค้าได้มากกว่า 1 จุด ทำให้สามารถลดจำนวนรอบการวิ่งได้น้อยกว่า โดยหลักการในการจัดเส้นทางของรถในแต่ละรอบการวิ่งเพื่อไปรับสินค้าในแต่ละจุดนั้นจะพิจารณาจากคู่ของลูกค้าที่มีการรวมรอบการวิ่งแล้วมีผลประหยัดมากที่สุดก่อนตามลำดับ โดยการรวมรอบการวิ่งนั้นไม่เกินเงื่อนไขของความสามารถในการบรรทุกและกรอบเวลาในการขนส่งของรถในรอบการวิ่งนั้นๆ



รูปที่ 2-3 การเปรียบเทียบการเดินทางแบบสองรอบและหนึ่งรอบ

$$S_{ij} = D_{0i} + D_{j0} - D_{ij} \qquad (10)$$

โดย $S_{ij} =$ ระยะทางการเดินทางที่ประหยัดได้เมื่อวิ่งรถรอบเดียว

$D_{0i} =$ ระยะทางการเดินทางจากคลังสินค้า (Depot) ไปยังลูกค้า i

$D_{j0} =$ ระยะทางการเดินทางจากลูกค้า j กลับมาถึงคลังสินค้า (Depot)

$D_{ij} =$ ระยะทางการเดินทางจากลูกค้า i ไปยังลูกค้า j

เมื่อ $i = 1,2,\ldots,N$

$j = 1,2,\ldots,N$

โดยที่ $i \neq j$

โดยขั้นตอนในการจัดเส้นทางโดยใช้ Saving Algorithms มีขั้นตอนดังต่อไปนี้

1. สร้างเมตริกระยะทาง (Distance Matrix) ระหว่างลูกค้าทั้งหมด
2. นำระยะทางของแต่ละคู่ของลูกค้ามาคำนวณ โดยการสร้างเมตริกแบบประหยัด (Saving Matrix) เพื่อหาค่าความประหยัดของแต่ละคู่ของลูกค้า
3. เริ่มทำการจัดเรียงเส้นทางโดยเรียงตามจากค่าความประหยัดที่ได้จากมากไปยังน้อยตามลำดับ
4. ถ้าจำนวนสินค้าทั้งหมดของทั้งสองเส้นทางไม่เกินความสามารถในการบรรทุกของรถและไม่เกินกรอบเวลาในการขนส่ง รอบการวิ่งทั้งสองจะถูกนำมารวมเป็นรอบการวิ่งเดียวกัน
5. ถ้าไม่สามารถเพิ่มจำนวนลูกค้าเข้ามาในเส้นทางได้อันเนื่องจาก เช่น ความสามารถในการบรรทุกของรถไม่เพียงพอ หรือ ข้อจำกัดด้านกรอบเวลาในการขนส่ง เป็นต้น จึงเริ่มจัดเส้นทางใหม่
6. ทำวนซ้ำไปเรื่อยๆจนครบทุกค่าความประหยัด จึงหยุดกระบวนการ

2.1.3 วิธีการเชิงพันธุกรรม (Genetic Algorithm) [6]

เป็นวิธีการเมตาฮิวริสติกที่จำลองการสืบพันธุกรรมของสิ่งมีชีวิต ซึ่งได้ถูกนำมาพัฒนาโดย Holland โดยใช้หลักการในการเรียนแบบการพัฒนาทางวิวัฒนาการโดยมีรากฐานมากทฤษฎีการวิวัฒนาการของ Chrales Dawin มาเป็นวิธีการในการคัดเลือกคำตอบที่ดี และมีการพัฒนาจากรุ่นสู่รุ่นเพื่อที่จะให้ได้คำตอบที่ดีที่สุด โดยนิยมใช้ในการจัดการปัญหาที่มีเงื่อนไขหลากหลายเข้ามาเกี่ยวข้อง

โดยลักษณะของรุ่นพ่อแม่จะถูกถ่ายทอดไปยังรุ่นลูกโดยใช้รูปแบบของโครโมโซม (Chromosome) และในโครโมโซมจะประกอบไปด้วยยีนส์ (Gene) ซึ่งจะบ่งบอกถึงลักษณะของสิ่งมีชีวิตนั้นๆ เช่น สีผิว สีผม เป็นต้น และจากวิธีการ Evolutionary Algorithm สิ่งมีชีวิตที่แข็งแรงหรือสามารถปรับตัวให้เข้ากับสภาพแวดล้อมได้จะมีโอกาสที่จะถูกสืบสายพันธุ์ต่อไป นอกจากนั้นในบางครั้งโครโมโซมลูกที่เกิดจากการคัดลอกของยีนส์พ่อและแม่อาจจะเกิดความไม่สมบรูณ์ทำให้เกิดการผ่าเหล่าหรือการปรับเปลี่ยนยีนส์ภายในโครโมโซม (Mutation) ในรุ่นลูกซึ่งมีโอกาสที่สามารถพัฒนาสายพันธุ์ที่ดียิ่งขึ้นได้ แต่ในทางตรงกันข้ามถ้าเกิดสายพันธุ์ที่ด้อยกว่า สายพันธุ์นั้นก็จะถูกกำจัดออกไปจากกระบวนการ

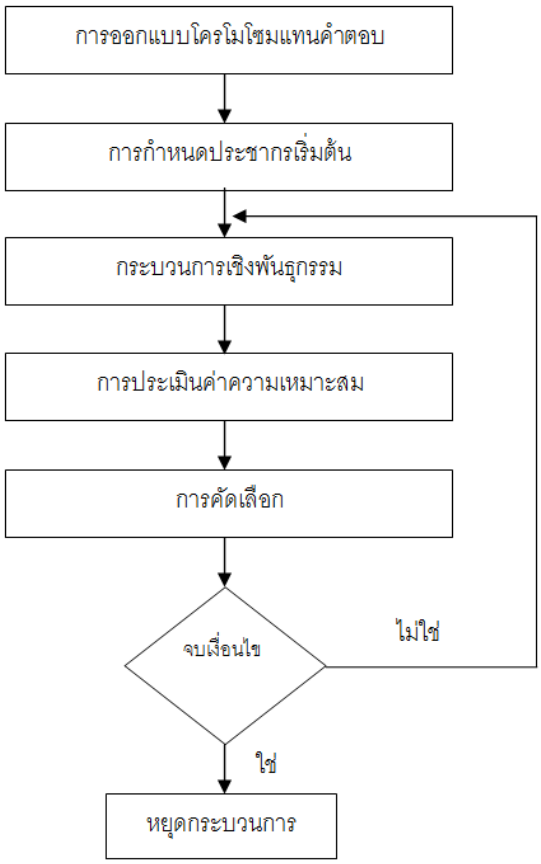
ตารางที่ 2-1 ความหมายของคำศัพท์ทางพันธุกรรม

คำศัพท์ทางพันธุกรรม	ความหมาย
โครโมโซม (Chromosome)	สตริงของผลลัพธ์ในรูปแบบของตัวแปรต่างๆ
ยีนส์ (Gene)	ตำแหน่งของตัวแปรต่างๆบนชุดสตริง
จำนวนประชากร (Population)	จำนวนผลลัพธ์สำหรับการพัฒนาหาค่าที่ดีที่สุดในแต่ละรุ่น
จำนวนรุ่นประชากร (Generation)	จำนวนรอบของการพัฒนาเพื่อค้นหาผลลัพธ์ที่ดีที่สุด
การคัดเลือกโดยธรรมชาติ (Natural Selection)	เป็นการประเมินความเหมาะสมของโครโมโซมโดยการกำหนดฟังก์ชั่นทางคณิตศาสตร์ เช่น การคัดเลือกแบบ Tournament , Roulette wheel selection , Elitise เป็นต้น
การแลกเปลี่ยนยีนส์ข้ามโครโมโซม (Crossover)	เป็นการแลกเปลี่ยนยีนส์ข้ามโครโมโซมระหว่างโครโมโซมพ่อและแม่ โดยการกำหนดจุดแบ่งหรือช่วงของยีนส์ในโครโมโซมพ่อและแม่และทำการสลับ เพื่อสร้างโครโมโซมรุ่นลูก
การแลกเปลี่ยนยีนส์ภายในโครโมโซม หรือ การผ่าเหล่า (Mutation)	เป็นการแลกเปลี่ยนยีนส์ภายในของโครโมโซมรุ่นลูก เพื่อหลีกเลี่ยงผลลัพธ์ที่เป็นค่าท้องถิ่น (Local optimal solution)

Genetic Algorithm มีลำดับขั้นตอนการทำงานหลักโดยทั่วไปประกอบด้วย 6 กระบวนการหลักๆ [7] ดังนี้

1. การออกแบบโครโมโซมแทนคำตอบ (Chromosome Encoding)
2. การกำหนดประชากรเริ่มต้น (Population Initialization)
3. กระบวนการเชิงพันธุกรรม (Genetic Operations)
4. การประเมินค่าความเหมาะสม (Fitness Evaluation)
5. การคัดเลือก (Selection)
6. การหยุดกระบวนการ (Stop Criteria)

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 2-4 ลำดับขั้นตอนวิธีการเชิงพันธุกรรม

2.1.3.1 การออกแบบโครโมโซมแทนคำตอบ (Chromosome Encoding)

การออกแบบโครโมโซมแทนคำตอบ เป็นขั้นตอนการแปลงผลลัพธ์ของปัญหาทางคณิตศาสตร์ให้อยู่ในรูปแบบของโครโมโซม ให้มีความสอดคล้องกับปัญหาจริงที่ต้องการหาคำตอบ วิธีการออกแบบโครโมโซมเพื่อแทนคำตอบนั้นสามารถทำได้หลายวิธี เช่น

- การออกแบบโครโมโซมแบบไบนารี (Binary Encoding) เป็นการออกแบบโครโมโซมที่เหมาะสำหรับการคำนวณหาค่าที่สูงสุด (Maximum) หรือค่าต่ำสุด (Minimum) ซึ่งแทนค่าด้วย 0 หรือ 1 เท่านั้น ตัวอย่างดังแสดงในรูปที่ 2-5

ลูกค้าจุดที่	1	2	3	4	5	6
โครโมโซม	1	0	1	1	0	1

รูปที่ 2-5 ค่าโครโมโซมของค่าความต้องการสินค้า

จากรูปที่ 2-5 สมมติว่าแทนค่า 1 คือมีความต้องการสินค้า และ 0 คือไม่มีความต้องการสินค้า โครโมโซมที่ได้คือ 101101 โดยแต่ละส่วนภายในโครโมโซมเรียกว่า ยีนส์ (Gene) เช่น ยีนที่1 มีค่าเท่ากับ 1 และยีนที่2 มีค่าเท่ากับ 0 เป็นต้น จะเห็นได้ว่าลูกค้าที่มี

ความต้องการสินค้า คือ ลูกค้าลำดับที่ 1 3 4 และ 6 และลูกค้าที่ไม่มีความต้องการสินค้า คือ ลูกค้าลำดับที่ 2 และ 5 เป็นต้น

- การออกแบบโครโมโซมแบบลำดับ (Permutation Encoding) เป็นการออกแบบโครโมโซมที่ใช้ตัวเลขได้ ซึ่งตัวเลขในแต่ละตัวแสดงถึงลำดับขั้นตอนในการทำงานหรือลำดับในการเดินทางขนส่ง ตัวอย่างดังแสดงในรูปที่ 2-6

ลำดับที่	1	2	3	4	5	6
โครโมโซม	5	6	3	1	2	4

รูปที่ 2-6 การออกแบบโครโมโซมแบบลำดับ

จากรูปที่ 2-6 แสดงให้เห็นว่ามีเมืองที่ต้องเดินทางขนส่งทั้งหมด 6 เมือง โดยเมืองที่เดินทางผ่านเป็นที่แรก คือ เมืองที่ 5 และเดินทางต่อไปยังเมืองที่ 6 3 1 2 และ 4 ตามลำดับ

- การออกแบบโครโมโซมแบบใช้ค่าหรือเครื่องหมายจริง (Value Encoding) เป็นการออกแบบโครโมโซมที่ใช้เลขจำนวนจริง ใช้ตัวอักษร หรือ กลุ่มวัตถุ เป็นตัวแทนเพื่อแสดงคำตอบมาใช้ในการแทนค่าในโครโมโซม ตัวอย่างดังแสดงในรูปที่ 2-7

สัปดาห์ที่	1	2	3	4	5	6
โครโมโซม	120	30	0	60	100	80

รูปที่ 2-7 การออกแบบโครโมโซมแบบใช้ค่าหรือเครื่องหมายจริง

จากรูปที่ 2-7 สมมติว่าเป็นปัญหาในการวางแผนการผลิตแสดงให้เห็นว่า สัปดาห์ที่1 มีการผลิต 120 หน่วย และสัปดาห์ที่ 2 มีการผลิต 30 หน่วย เป็นต้น

2.1.3.2 การกำหนดประชากรเริ่มต้น (Population Initialization)

เป็นการเลือกประชากรเริ่มต้นเพื่อนำไปสู่การพัฒนาของ Genetic Algorithm โดยประชากรเริ่มต้นเกิดจากการสุ่มหรือคัดเลือกจากวิธีต่างๆ เพื่อให้ได้ประชากรเริ่มต้นจำนวนหนึ่ง และจำนวนประชากรเริ่มต้นที่ได้มานั้นเป็นพารามิเตอร์ที่ต้องกำหนดขึ้นมาก่อนที่จะเริ่มเข้าสู่ Genetic Algorithm โดยการกำหนดประชากรเริ่มต้นนั้นสามารถทำได้หลายวิธี เช่น

- การสุ่มแบบมีความน่าจะเป็นเท่ากันทุกกรณี (Random) จากปัญหาการหาขนาดการผลิตในแต่ละเดือน ถ้าตัดสินใจว่าจะผลิต (ยีนส์เท่ากับ1) หรือถ้าไม่ผลิต (ยีนส์เท่ากับ 0) ตัวอย่างดังแสดงในรูปที่ 2-8

เดือน	ตัวเลขสุ่ม	ผลการสุ่ม
1	0.55	0
2	0.37	1
3	078	0
4	0.25	1

รูปที่ 2-8 การสุ่มเลือกประชากรตั้งต้น

จากรูปที่ 2-8 ในแต่ละเดือนยีนส์ที่เป็นไปได้ คือ 0 กับ 1 ซึ่งมีความน่าจะเป็นเท่ากัน จากการสุ่มหากสุ่มได้เลขที่น้อยกว่าหรือเท่ากับ 0.5 จะเลือกผลิต เช่น สัปดาห์ที่ 2 และ 4 และหากสุ่มได้เลขที่มากกว่า 0.5 จะไม่ทำการผลิต เช่น สัปดาห์ที่ 1 และ 3 เป็นต้น

- การสุ่มแบบมีความน่าจะเป็นไม่เท่ากัน (Greedy Random) การสุ่มแบบมีความน่าจะเป็นไม่เท่ากันนั้นแต่ละทางเลือกจะมีความน่าจะเป็นที่ไม่เท่ากัน ขึ้นอยู่กับค่าตัวแปรบางตัวในปัญหานั้นๆ

2.1.3.3 กระบวนการเชิงพันธุกรรม (Genetic Operations)

กระบวนการเชิงพันธุกรรมมีขั้นตอนหลักๆอยู่ 2 ขั้นตอน คือ การแลกเปลี่ยนยีนส์ข้ามโครโมโซม (Crossover) และการปรับเปลี่ยนยีนส์ภายในโครโมโซม (Mutation) ในกระบวนการนี้จะมีการค้นหาและพัฒนาผลคำตอบเพื่อหาผลเฉลยที่ดีขึ้น

- การแลกเปลี่ยนยีนส์ข้ามโครโมโซม (Crossover) เป็นขั้นตอนที่ทั้งโครโมโซมพ่อและแม่ (Parents) ดำเนินการแลกเปลี่ยนยีนส์ระหว่างกันเพื่อสร้างโครโมโซมชุดใหม่ คือ โครโมโซมลูก (Offspring) โดยทั่วไปการแลกเปลี่ยนยีนส์ข้ามโครโมโซมมีความน่าจะเป็น (Crossover Probability) ในการเกิดประมาณ 0.6 – 0.9 โดยวิธีการแลกเปลี่ยนยีนส์ข้ามโครโมโซมนี้มีหลายวิธี ดังนี้

1. การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบหนึ่งจุด (Single Point Crossover)
2. การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด (Two Point Crossover)
3. การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ Partial Mapping Crossover (PMX)
4. การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ Weight Mapping Crossover (WMX)

- การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบหนึ่งจุด (Single Point Crossover)

โดยการนำโครโมโซมของพ่อและแม่ มาทำการสุ่มตำแหน่งจุดตัดจำนวน 1 จุด ซึ่งจุดตัดของโครโมโซมพ่อและแม่ต้องตรงกัน แล้วจึงตัดโครโมโซมออกเป็น 2 ส่วน ตัวอย่างดังแสดงในรูปที่ 2-9

โครโมโซมของพ่อ (P_1)	(3 4 | 2 6 1 5)
โครโมโซมของแม่ (P_2)	(6 5 | 4 1 3 2)

รูปที่ 2-9 การกำหนดจุดตัดการแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบหนึ่งจุด

ทำการถ่ายทอดยีนส์ในช่วงแรกที่ทำการตัดจากโครโมโซมของพ่อ (P_1)และแม่ (P_2) ไปยังโครโมโซมของลูกคนที่1 (O_1) และลูกคนที่2 (O_2) ตามลำดับ และแทนค่าด้วยเครื่องหมาย x ในช่วงหลังที่ยังไม่ได้รับการถ่ายทอดยีนส์ ตัวอย่างดังแสดงในรูปที่ 2-10

โครโมโซมของลูกคนที่1 (O_1)	(3 4 | x x x x)
โครโมโซมของลูกคนที่2 (O_2)	(6 5 | x x x x)

รูปที่ 2-10 การถ่ายทอดยีนส์ที่ถูกตัดในช่วงแรกมายังโครโมโซมลูก

เมื่อได้ถ่ายทอดยีนส์ในช่วงแรกเสร็จแล้วจากนั้นพิจารณายีนส์จากพ่อและแม่ โดยทำการตัดยีนส์ที่มีค่าซ้ำกับในส่วนที่ลูกรับได้การถ่ายถอดแล้วออกไป ตัวอย่างดังแสดงในรูปที่ 2-11

โครโมโซมของพ่อ (P_1)	(3 4 2 x 1 x)
โครโมโซมของแม่ (P_2)	(6 5 x 1 x 2)

รูปที่ 2-11 การแลกเปลี่ยนยีนข้ามโครโมโซมแบบหนึ่งจุด โดยทำการตัดตัวที่ซ้ำออก

ทำการถ่ายทอดยีนส์จากพ่อและแม่ที่ได้ทำการตัดค่าที่ซ้ำออกไปแล้วไปยังรุ่นลูกในส่วนที่ยังไม่ได้รับการถ่ายทอด โดยโครโมโซมพ่อ (P_1) ไปยังโครโมโซมลูกคนที่2 (O_2) และโครโมโซมแม่ (P_2) ไปยังโครโมโซมลูกที่1 (O_1) ตัวอย่างดังแสดงในรูปที่ 2-12

โครโมโซมของลูกคนที่1 (O_1)	(3 4 | 6 5 1 2)
โครโมโซมของลูกคนที่2 (O_2)	(6 5 | 3 4 2 1)

รูปที่ 2-12 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบหนึ่งจุด โดยทำการถ่ายทอดยีนส์ที่ถูกตัดตัวซ้ำแล้ว

โครโมโซมของพ่อ (P_1)	*3 4* 2 6 1 5
โครโมโซมของแม่ (P_2)	*6 5* 4 1 3 2
โครโมโซมของลูกคนที่1 (O_1)	*3 4* 6 5 1 2
โครโมโซมของลูกคนที่2 (O_2)	*6 5* 3 4 2 1

รูปที่ 2-13 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบหนึ่งจุด

- การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด (Two Point Crossover)

โดยการนำโครโมโซมของพ่อและแม่ มาทำการสุ่มตำแหน่งจุดตัดจำนวน 2 จุด ซึ่งจุดตัดของโครโมโซมพ่อและแม่ต้องตรงกัน ตัวอย่างดังแสดงในรูปที่ 2-14

โครโมโซมของพ่อ (P_1)	(2 I 6 4 5 1 I 3)
โครโมโซมของแม่ (P_2)	(5 I 2 6 1 3 I 4)

รูปที่ 2-14 การกำหนดจุดตัดการแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด

ทำการถ่ายทอดยีนส์ในช่วงที่ทำการตัดจากโครโมโซมของพ่อ (P_1)และแม่ (P_2) ไปยังโครโมโซมของลูกคนที่1 (O_1) และลูกคนที่2 (O_2) ตามลำดับ และแทนค่าด้วยเครื่องหมาย x ในช่วงที่ยังไม่ได้รับการถ่ายทอดยีนส์ ตัวอย่างดังแสดงในรูปที่ 2-15

โครโมโซมของลูกคนที่1 (O_1)	(2 I x x x x I 3)
โครโมโซมของลูกคนที่2 (O_2)	(5 I x x x x I 4)

รูปที่ 2-15 การถ่ายทอดยีนส์ที่ถูกตัดในช่วงมายังโครโมโซมลูก

เมื่อได้ถ่ายทอดยีนส์ในช่วงเสร็จแล้วพิจารณายีนส์จากพ่อและแม่ โดยทำการตัดยีนส์ที่มีค่าซ้ำในส่วนที่ลูกรับได้รับการถ่ายถอดไปแล้วออกไป ตัวอย่างดังแสดงในรูปที่ 2-16

โครโมโซมของพ่อ (P_1)	(2 6 x x 1 3)
โครโมโซมของแม่ (P_2)	(5 x 6 1 x 4)

รูปที่ 2-16 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด โดยทำการตัดตัวที่ซ้ำออก

ทำการถ่ายทอดยีนส์จากพ่อและแม่ที่ได้ทำการตัดค่าที่ซ้ำออกไปแล้วไปยังรุ่นลูกในส่วนที่ยังไม่ได้รับการถ่ายทอด โดยโครโมโซมพ่อ (P_1) ไปยังโครโมโซมลูกคนที่2 (O_2) และโครโมโซมแม่ (P_2) ไปยังโครโมโซมลูกที่1 (O_1) ตัวอย่างดังแสดงในรูปที่ 2-17

โครโมโซมของลูกคนที่1 (O_1)	(2 5 6 1 4 3)
โครโมโซมของลูกคนที่2 (O_2)	(5 2 6 1 3 4)

รูปที่ 2-17 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด โดยทำการถ่ายทอดยีนส์ที่ถูกตัดตัวซ้ำแล้ว

โครโมโซมของพ่อ (P_1)	*2* 6 4 5 1 *3*
โครโมโซมของแม่ (P_2)	*5* 2 6 1 3 *4*
โครโมโซมของลูกคนที่1 (O_1)	*2* 5 6 1 4 *3*
โครโมโซมของลูกคนที่2 (O_2)	*5* 2 6 1 3 *4*

รูปที่ 2-18 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบสองจุด

- การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ Partial Mapping Crossover (PMX)

เริ่มจากการสุ่มเลือกจุดตัดมาทั้งหมด 2 จุด โดยจุดตัดของทั้งโครโมโซมพ่อและโครโมโซมแม่ที่ทำการสุ่มขึ้นมาต้องมีจุดตัดที่ตรงกันด้วย ตัวอย่างดังแสดงในรูปที่ 2-19

โครโมโซมของพ่อ (P_1)	(5 3 I 6 1 4 I 2)
โครโมโซมของแม่ (P_2)	(4 2 I 1 6 5 I 3)

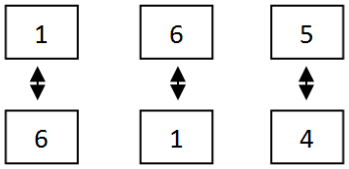
รูปที่ 2-19 การกำหนดจุดตัดการแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX

จากนั้นทำการสลับยีนส์ที่อยู่ในช่วงของจุดตัดทั้ง 2 จุดของโครโมโซมของพ่อ (P_1) และโครโมโซมของแม่ (P_2) เพื่อสร้างโครโมโซมรุ่นลูก ตัวอย่างดังแสดงในรูปที่ 2-20

โครโมโซมของลูกคนที่1 (O_1)	(5 3 I 1 6 5 I 2)
โครโมโซมของลูกคนที่2 (O_2)	(4 2 I 6 1 4 I 3)

รูปที่ 2-20 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX โดยทำการสลับยีนส์ข้ามโครโมโซมของช่วงที่ถูกตัด

พิจารณาช่วงของยีนส์ที่อยู่ในช่วงของจุดตัดทั้ง 2 จุด ของโครโมโซมของพ่อ (P_1) และโครโมโซมของแม่ (P_2) ตัวอย่างดังแสดงในรูปที่ 2-21



รูปที่ 2-21 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX โดยทำการสร้างแผนภาพความสัมพันธ์

จากนั้นสังเกตที่โครโมโซมลูกคนที่1 (O_1) และโครโมโซมลูกคนที่2 (O_2) ถ้ายีนส์ที่ไม่ได้อยู่ในช่วงที่ถูกตัดมีการซ้ำกับยีนส์ที่อยู่ในช่วง ให้ทำการเปลี่ยนค่ายีนส์ตามแผนภาพความสัมพันธ์ ตัวอย่างดังแสดงในรูปที่ 2-22

โครโมโซมของลูกคนที่1 (O_1)	(4 3 I 1 6 5 I 2)
โครโมโซมของลูกคนที่2 (O_2)	(5 2 I 6 1 4 I 3)

รูปที่ 2-22 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX โดยทำการเปลี่ยนค่ายีนส์จากแผนภาพความสัมพันธ์

โครโมโซมของพ่อ (P_1)	(5 3 *6* *1* *4* 2)
โครโมโซมของแม่ (P_2)	(4 2 *1* *6* *5* 3)
โครโมโซมของลูกคนที่1 (O_1)	(4 3 *1* *6* *5* 2)
โครโมโซมของลูกคนที่2 (O_2)	(5 2 *6* *1* *4* 3)

รูปที่ 2-23 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ PMX

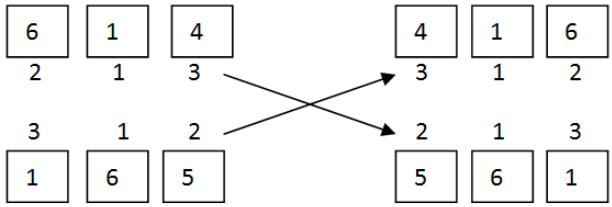
- การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ Weight Mapping Crossover (WMX)

เริ่มจากการสุ่มเลือกจุดตัดมาทั้งหมด 2 จุด โดยจุดตัดของทั้งโครโมโซมพ่อและโครโมโซมแม่ที่ทำการสุ่มขึ้นมาต้องมีจุดตัดที่ตรงกันด้วย ตัวอย่างดังแสดงในรูปที่ 2-24

โครโมโซมของพ่อ (P_1)	(5 3 I 6 1 4 I 2)
โครโมโซมของแม่ (P_2)	(4 2 I 1 6 5 I 3)

รูปที่ 2-24 การกำหนดจุดตัดการแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ WMX

จากนั้นทำการกำหนดค่าน้ำหนักโดยการสุ่มให้กับยีนส์ที่อยู่ในช่วงของจุดตัดทั้ง 2 จุดของโครโมโซมของพ่อ (P_1) และโครโมโซมของแม่ (P_2) และทำการสลับค่าน้ำหนักระหว่างกัน จากนั้นจึงทำเปลี่ยนค่าของยีนส์เพื่อสร้างโครโมโซมรุ่นลูก ตัวอย่างดังแสดงในรูปที่ 2-25



รูปที่ 2-25 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ WMX โดยทำการสร้างแผนภาพความสัมพันธ์

จากนั้นจะได้โครโมโซมลูกคนที่1 (O_1) และโครโมโซมลูกคนที่2 (O_2) ที่ได้ทำการสลับตำแหน่งของยีนส์โดยการกำหนดค่าน้ำหนักโดยวิธีการสุ่มตามแผนภาพความสัมพันธ์ ตัวอย่างดังแสดงในรูปที่ 2-26

โครโมโซมของลูกคนที่1 (O_1)	(5 3 I 4 1 6 I 2)
โครโมโซมของลูกคนที่2 (O_2)	(4 2 I 5 6 1 I 3)

รูปที่ 2-26 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ WMX โดยทำการเปลี่ยนค่ายีนส์จากแผนภาพความสัมพันธ์

โครโมโซมของพ่อ (P_1)	(5 3 *6* *1* *4* 2)
โครโมโซมของแม่ (P_2)	(4 2 *1* *6* *5* 3)
โครโมโซมของลูกคนที่1 (O_1)	(5 3 *4* *1* *6* 2)
โครโมโซมของลูกคนที่2 (O_2)	(4 2 *5* *6* *1* 3)

รูปที่ 2-27 การแลกเปลี่ยนยีนส์ข้ามโครโมโซมแบบ WMX

- การปรับเปลี่ยนยีนส์ภายในโครโมโซม (Mutation) เป็นขั้นตอนถัดไปจากการครอสโอเวอร์โดยโครโมโซมแต่ละตัวจะทำการแลกเปลี่ยนยีนส์ภายในของแต่ละโครโมโซม โดยทั่วไปการแลกเปลี่ยนยีนส์ภายในโครโมโซมมีความน่าจะเป็น (Mutation Probability) ในการเกิดประมาณ 0.1 – 0.3 โดยวิธีการแลกเปลี่ยนยีนส์ภายในโครโมโซมนี้มีหลายวิธี ดังนี้

1. การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก (Insertion Mutation)
2. การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง (Displacement Mutation)
3. การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง (Reciprocal Exchange Mutation)

- การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก (Insertion Mutation)

เริ่มต้นโดยสุ่มเลือกโครโมโซมจากประชากรทั้งหมดขึ้นมา 1 ตัว แล้วนำมาเป็นโครโมโซมในรุ่นพ่อและแม่ ตัวอย่างดังแสดงในรูปที่ 2-28

โครโมโซมของพ่อและแม่ (P_1)	(5 2 4 6 1 3)

รูปที่ 2-28 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก ขั้นตอนการเลือกโครโมโซม

ทำการสุ่มตำแหน่งของยีนส์ที่ต้องการจะทำการแทรกลงไป ตัวอย่างดังแสดงในรูปที่ 2-29

โครโมโซมของพ่อและแม่ (P_1)	(5 2 4 6 1 3)

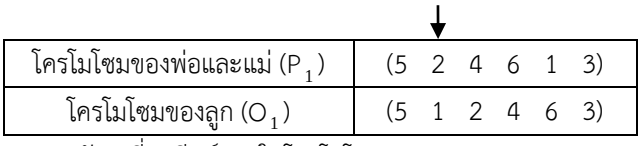
รูปที่ 2-29 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก ขั้นตอนการสุ่มตำแหน่งยีนส์ที่จะทำการแทรก

ทำการสุ่มเลือกชุดของยีนส์ที่จะนำเข้ามาแทรก ตัวอย่างดังแสดงในรูปที่ 2-30

โครโมโซมของพ่อและแม่ (P_1)	(5 2 4 6 I 1 I 3)

รูปที่ 2-30 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก ขั้นตอนการสุ่มเลือกยีนส์ที่จะนำเข้ามาแทรก

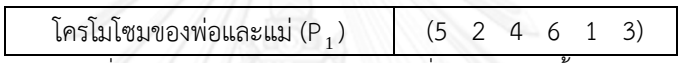
นำค่าของชุดยีนส์ที่ทำการเลือกไว้ไปแทรกแทนที่ตำแหน่งที่ได้ทำการสุ่มไว้ก่อนหน้านี้ นั่นคือ นำค่ายีนส์ 1 ไปแทรกลงในตำแหน่งที่ 2 ของโครโมโซม ตัวอย่างดังแสดงในรูปที่ 2-31



รูปที่ 2-31 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบแทรก (Insertion Mutation)

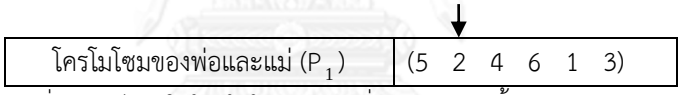
- การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง (Displacement Mutation)

เริ่มต้นโดยการสุ่มเลือกโครโมโซมจากประชากรทั้งหมดขึ้นมา 1 ตัว แล้วนำมาเป็นโครโมโซมในรุ่นพ่อและแม่ ตัวอย่างดังแสดงในรูปที่ 2-32



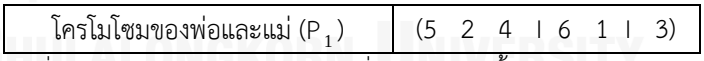
รูปที่ 2-32 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง ขั้นตอนการสุ่มเลือกโครโมโซม

ทำการสุ่มตำแหน่งของยีนส์ที่ต้องการจะทำการเคลื่อนตำแหน่งลงไป ตัวอย่างดังแสดงในรูปที่ 2-33



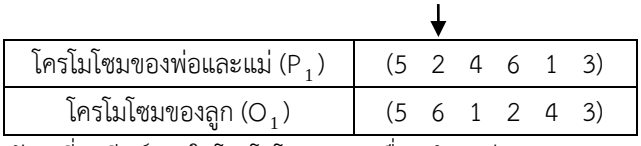
รูปที่ 2-33 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง ขั้นตอนการสุ่มตำแหน่งยีนส์ที่จะทำการเคลื่อนตำแหน่ง

ทำการสุ่มเลือกช่วงชุดของยีนส์ที่จะนำเข้ามาเคลื่อนตำแหน่ง ตัวอย่างดังแสดงในรูปที่ 2-34



รูปที่ 2-34 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง ขั้นตอนการสุ่มเลือกช่วงยีนส์ที่จะนำเข้ามาเคลื่อนตำแหน่ง

นำค่าของช่วงชุดยีนส์ที่ทำการเลือกไว้ไปแทรกแทนที่ตำแหน่งที่ได้ทำการสุ่มไว้ก่อนหน้านี้ นั่นคือ นำค่ายีนส์ 6 และ 1 ไปแทรกลงในตำแหน่งที่ 2 ของโครโมโซม ตัวอย่างดังแสดงในรูปที่ 2-35



รูปที่ 2-35 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบเคลื่อนตำแหน่ง (Displacement Mutation)

- การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง (Reciprocal Exchange Mutation)

เริ่มต้นโดยการสุ่มเลือกโครโมโซมจากประชากรทั้งหมดขึ้นมา 1 ตัว แล้วนำมาเป็นโครโมโซมในรุ่นพ่อและแม่ ตัวอย่างดังแสดงในรูปที่ 2-36

โครโมโซมของพ่อและแม่ (P_1)	(1 4 6 3 5 2)

รูปที่ 2-36 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง ขั้นตอนการสุ่มเลือกโครโมโซม

ทำการสุ่มตำแหน่งของยีนส์ที่ต้องการจะทำการสลับตำแหน่งขึ้นมา 2 ตำแหน่ง ตัวอย่างดังแสดงในรูปที่ 2-37

โครโมโซมของพ่อและแม่ (P_1)	(1 4 6 3 5 2)

รูปที่ 2-37 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง ขั้นตอนการสุ่มตำแหน่งยีนส์

ทำการสลับตำแหน่งของยีนส์ทั้ง 2 ตำแหน่ง ที่ได้จากการสุ่มเลือกมาเพื่อสร้างโครโมโซมรุ่นลูก ตัวอย่างดังแสดงในรูปที่ 2-38

โครโมโซมของพ่อและแม่ (P_1)	(1 4 5 3 6 2)

รูปที่ 2-38 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง ขั้นตอนการสลับตำแหน่งยีนส์

โครโมโซมของของพ่อและแม่ (P_1)	(1 4 *6* 3 *5* 2)
โครโมโซมของลูก (O_1)	(1 4 *5* 3 *6* 2)

รูปที่ 2-39 การปรับเปลี่ยนยีนส์ภายในโครโมโซมแบบสลับตำแหน่ง

2.1.3.4 การประเมินค่าความเหมาะสม (Fitness Evaluation)

เป็นการพิจารณาความเหมาะสมของผลลัพธ์ในฟังก์ชั่นจากสมการแทนค่าคำตอบ (Fitness Function) ซึ่งผลการประเมินค่าความเหมาะสมมีที่มาจาก 2 ประเภท คือ มาจากคำตอบของฟังก์ชั่นที่ต้องการแก้ปัญหา หรือ มาจากฟังก์ชั่นอื่นๆ

2.1.3.5 การคัดเลือก (Selection)

เป็นการคัดเลือกโครโมโซมจากสมาชิกทั้งหมดที่มีความเหมาะสมของผลลัพธ์จาก Genetic Algorithm เพื่อเป็นโครโมโซมตั้งต้นที่ดีในรอบกระบวนการถัดไป โดยโครโมโซมที่มีความเหมาะสม (Fitness) ที่มากกว่าจะมีโอกาสถูกคัดเลือกไปในรอบถัดไปมากกว่า โดยวิธีการคัดเลือกมีหลากหลายวิธี เช่น

- การคัดเลือกแบบใช้วงล้อสัดส่วนค่าความเหมาะสม (Roulette wheel selection)

การคัดเลือกวิธีนี้ใช้สัดส่วนค่าความเหมาะสม (Fitness) ของแต่ละโครโมโซมในการคัดเลือก โดยใช้การแบ่งขนาดพื้นที่ของวงล้อความเหมาะสม โครโมโซมที่เหมาะสมจะมีค่า Fitness ที่มากทำให้มีพื้นที่ในวงล้อสัดส่วนที่มากกว่าโครโมโซมที่ด้อย เมื่อมีการหมุนวงล้อสัดส่วนโครโมโซมที่มีความเหมาะสมมากกว่าจะมีโอกาสถูกคัดเลือกมากกว่าด้วย ตามสมการที่ 11 ดังนี้

$$P_i = \frac{F_i}{\sum_{n=1}^{N} F_i} \qquad \text{เมื่อ} \quad i = 1,2,..,N \tag{11}$$

โดย $P_i =$ โอกาสของโครโมโซม i ที่จะถูกคัดเลือก

$F_i =$ ค่าความเหมาะสมของโครโมโซม i

$N =$ ประชากรทั้งหมด

- การคัดเลือกแบบใช้การแข่งขัน (Tournament selection)

การคัดเลือกวิธีนี้ใช้การเปรียบเทียบค่าความเหมาะสม (Fitness) ระหว่างหลายๆ โครโมโซม โดยการเลือกจะแบ่งการคัดเลือกออกเป็นเซตๆ และทำการสุ่มเลือกโครโมโซมมาทำการแข่งขันเพื่อหาโครโมโซมที่ดีที่สุดในแต่ละเซต ซึ่งจำนวนเซตหมายถึงจำนวนประชากรที่จะถูกคัดเลือก

- การคัดเลือกแบบใช้การจัดลำดับ (Ranking selection)

การคัดเลือกวิธีนี้ใช้การเปรียบเทียบค่าความเหมาะสม (Fitness) ของทุกๆโครโมโซม โดยเลือกโครโมโซมที่มีค่าความเหมาะสมที่ดีที่สุดขึ้นก่อนและโครโมโซมที่มีค่าความเหมาะสมที่น้อยกว่าตามลำดับ

2.1.3.6 การหยุดกระบวนการ (Stop Criteria)

เป็นขั้นตอนการหยุดเพื่อจบกระบวนการ Genetic Algorithm โดยเงื่อนไขในกำหนดการหยุดของกระบวนการสามารถกำหนดได้หลากหลาย เช่น การกำหนดจำนวนรอบกระบวนการ เวลาที่ใช้ดำเนินการ หรือการกำหนดจำนวนรอบของผลลัพธ์ที่ซ้ำในแต่ละรอบกระบวนการ เป็นต้น

2.2 เอกสารและงานวิจัยที่เกี่ยวข้อง

ในส่วนของเอกสารและงานวิจัยที่เกี่ยวข้องจะกล่าวเกี่ยวกับ งานวิจัยต่างๆที่ได้นำวิธีทางฮิวริสติกมาประยุกต์ใช้ในงานวิจัยต่างๆ เพื่อทำการหาคำตอบ ดังนี้

อุบลรัตน์ เธียรธนาคม [8] ได้ประยุกต์ใช้วิธีการฮิวริสติกในการแก้ปัญหาการจัดเส้นทางยานพาหนะที่มีคลังสินค้าหลายแห่งโดยได้ประยุกต์ใช้ Saving Algorithm เพื่อหาคำตอบตั้งต้นและปรับปรุงคำตอบตั้งต้นด้วย Genetic Algorithm โดยมีการออกแบบโครโมโซมแทนคำตอบประกอบด้วย ข้อมูลคลังสินค้า ลำดับรอบการวิ่ง และลำดับของลูกค้าภายในลำดับรอบการวิ่ง โดยพบว่า Genetic Algorithm สามารถหาระยะทางขนส่งรวมได้น้อยลงกว่า Saving Algorithm โดยเฉลี่ย 20%

ศุภพัชร พวงแก้ว [9] ได้ใช้ Genetic Algorithm สำหรับปัญหาการจัดเส้นทางเดินรถเก็บขยะตามครัวเรือนโดยใช้คนงานในการจัดเก็บ โดยได้นำเสนอแบบจำลองทางคณิตศาสตร์ซึ่งไม่สามารถนำมาแก้ปัญหาในการจัดเส้นทางเดินของรถจัดเก็บขยะได้ เนื่องจากมีจำนวนจุดเก็บจำนวนมากทำเกิดความซับซ้อนในการจัดเส้นทางซึ่งส่งผลต่อเวลาที่ใช้ในการหาคำตอบที่มากจนบางทีได้เวลาที่ไม่เหมาะสม จึงได้มีการนำ Genetic Algorithm มาประยุกต์ใช้ในปัญหาการจัดเส้นทางเพื่อหาคำตอบภายในเวลาที่เหมาะสม โดยพารามิเตอร์ถูกกำหนดตามผลการทดลอง พบว่า Genetic Algorithm มีความสามารถในการหาคำตอบในทุกๆขนาดของปัญหาโดยใช้เวลาในการประมวลผลที่น้อยกว่า

ธฤต รัตนมณี [10] ได้ทำการการพัฒนาฮิวริสติกบนพื้นฐานของ Genetic Algorithm สำหรับการจัดเส้นทางรถขนส่งกรณีมีรถขนส่งหลายขนาดแบบหลายช่องบรรจุและหลายผลิตภัณฑ์ของการขนส่งน้ำมันเชื้อเพลิง ผู้วิจัยได้ทำการสร้างแบบจำลองทางคณิตศาสตร์ขึ้นมา เนื่องจากวิธีการหาคำตอบที่ดีที่สุดไม่สามารถหาได้ในกรณีที่มีปัญหาขนาดใหญ่ ผู้วิจัยจึงได้นำ Genetic Algorithm มาประยุกต์ใช้โดยได้ออกแบบโครโมโซมซึ่งแยกเป็นสองส่วนและพัฒนาคำตอบโดยวิธีการแลกเปลี่ยนสายพันธุ์และการกลายพันธุ์เพื่อใช้หาคำตอบที่ดีในเวลาการคำนวณที่เหมาะสมด้วย เมื่อเปรียบเทียบผล พบว่า Genetic Algorithm มีความสามารถในการหาคำตอบและใช้เวลาในการประมวลน้อยกว่ามาก

พศวีร์ ตรีวิเศษ [11] ได้ทำการเปรียบเทียบประสิทธิภาพของวิธีการจัดเส้นทางทั้งหมด 4 วิธี คือ วิธี 1.Greedy Algorithm 2.Saving Algorithm 3.Nearest Algorithm และ 4.Genetic Algorithm ในการขนส่งสินค้าวัสดุก่อสร้างไปถึงลูกค้า เพื่อหาเส้นทางการขนส่งที่สั้นที่สุด จากการเปรียบเทียบพบว่าวิธี Genetic Algorithm สามารถจัดเส้นทางการขนส่งได้สั้นที่สุด สามารถลดลงร้อยละ 24.38

วันวิสา นฤมิตวงศ์ [12] กล่าวถึงการตั้งค่าพารามิเตอร์ของ Genetic Algorithm ในการจัดลำดับการผลิตไว้โดยการทำ Weighted Mapping Crossoverและ Reciprocal Exchange Mutation โดยทดลองค่าความน่าจะเป็นในการครอสโอเวอร์ (Pc) และการมิวเตชั่น (Pm) อยู่ในช่วง 0.7-0.9 และ 0.1-0.3 ตามลำดับ และกำหนดโครโมโซมตั้งต้นในกระบวนการจำนวน 100 โครโมโซม ว่าเป็นค่าที่พารามิเตอร์ที่เหมาะสมที่สุดของลักษณะปัญหา

K.C.Tan et al. [13] กล่าวถึงการตั้งค่าพารามิเตอร์ของ Genetic Algorithm ของปัญหาการจัดเส้นทางโดยมีเงื่อนไขกรอบเวลาไว้ว่าควรจะตั้งโครโมโซมตั้งต้นในกระบวนการประมาณ 100 โครโมโซม และตั้งให้มีกระบวนการเชิงพันธุกรรมประมาณ 500 ถึง 1000 รอบ เพราะมีความเหมาะสมของเวลาในการประมวลผลและคำตอบสุดท้ายที่ได้

จากการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องจึงทำให้ผู้วิจัยตระหนักถึงความจำเป็นในการนำเอาวิธีการทางฮิวริสติกทั้ง Saving Algorithm และ Genetic Algorithm มาประยุกต์ใช้เพื่อสร้างเส้นทางการขนส่งของซากและเศษซากไก่ตายสำหรับโรงกำจัดซากไก่ให้มีประสิทธิภาพทั้งในด้านค่าใช้จ่ายและระยะทาง

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 3
วิธีการดำเนินงานวิจัย

ในวิธีการดำเนินงานวิจัยจะประกอบไปด้วยการเตรียมข้อมูลของงานวิจัย การสร้างแบบจำลองทางคณิตศาสตร์ และการพัฒนาวิธีการหาคำตอบตั้งต้นโดยวิธี Saving Algorithm และทำการปรับปรุงคำตอบตั้งต้นโดยวิธี Genetic Algorithm ซึ่งในบทนี้จะกล่าวถึงเฉพาะวิธีการหาคำตอบตั้งต้นเท่านั้น

3.1 การเตรียมข้อมูลในงานวิจัย

ในวิธีการดำเนินงานวิจัยเริ่มต้นผู้วิจัยได้ทำการเตรียมข้อมูลในงานวิจัย ซึ่งประกอบไปด้วยข้อมูลปริมาณซากไก่และเศษซากไก่ในศูนย์รวมแต่ละจุดของลูกค้าทั้ง 22 จุด ข้อมูลที่ตั้งของโรงกำจัดซากไก่และลูกค้าทั้งหมด และค่าใช้จ่ายในการขนส่งรวมไปถึงเงื่อนไขทางด้านความจุและกรอบเวลาในแต่ละรอบการวิ่ง จากนั้นจึงทำการสร้างแบบจำลองทางคณิตศาสตร์ (Mathematical model) ของปัญหาการจัดเส้นทางเดินรถแบบมีเงื่อนไขเวลาและความจุของรถมีจำกัด (Vehicle Routing Problem with Time Window and Capacitated Vehicle Constraints) และมีการพัฒนาวิธีการหาคำตอบตั้งต้นโดยวิธี Saving Algorithm และปรับปรุงเส้นทางจากคำตอบตั้งต้นโดย Genetic Algorihm โดยในงานวิจัยนี้มีสมมุติฐานที่ใช้ในการจัดเส้นทางการขนส่งของโรงกำจัดซากไก่ ดังนี้

1. การศึกษาข้อมูลเพื่อจัดเส้นทางการขนส่งของโรงกำจัดซากไก่
2. การวัดระยะทางการขนส่ง
3. ปริมาณสินค้าประเภทซากไก่
4. ปริมาณสินค้าประเภทเศษซากไก่
5. รถที่ใช้ในการขนส่ง
6. การคำนวณเวลาในการเดินทาง
7. การคำนวณเวลาในการขนสินค้า
8. การคิดค่าใช้จ่ายในการขนส่งต่อปี
9. การวิ่งของรถที่วิ่งไปรับสินค้าในแต่ละรอบการวิ่ง
10. การจัดรอบการวิ่งและลำดับเส้นทาง

3.1.1 การศึกษาข้อมูลเพื่อจัดเส้นทางการขนส่งของโรงกำจัดซากไก่

การศึกษาข้อมูลเพื่อจัดเส้นทางการขนส่งของโรงกำจัดซากไก่ คือ มีการตั้งโรงกำจัดซากไก่ 1 โรง (The Single Depot Vehicle Routing Problem) โดยมีตัวอย่างบริเวณสถานที่ตั้งของโรงกำจัดซากไก่ทั้งหมด 2 แห่ง คือ 1) ตำบลลำพญากลาง จังหวัดลพบุรี และ 2) ตำบลทับกวาง จังหวัดสระบุรี โดยโรงกำจัดซากไก่ที่เลือกนี้จะรับสินค้าทั้งหมด 2 ประเภท คือ 1) ซากไก่ตายจากฟาร์ม สำหรับป้อนสายการผลิตที่ 1 มีกำลังการผลิต 30,000 ตันต่อปี และ 2) เศษซากไก่จากโรงเชือด สำหรับป้อนสายการผลิต ที่ 2 มีกำลังการผลิต 30,000

ตันต่อปี ซึงเมื่อคำนวณปริมาณทั้งหมดแล้วพบว่าจะมีปริมาณเกินกำลังการผลิตของสายการผลิตทั้ง 2 สาย จึงปรับปริมาณวัตถุดิบที่จะไปรับทั้ง 22 จุด โดยการปรับตามอัตราส่วน (proportional) เพื่อให้สามารถรองรับกำลังการผลิตได้ โดยรวมจำนวนจุดที่ต้องมีการขนส่งสินค้าทั้งหมด 22 จุด ที่อยู่ในระยะทางที่พอเหมาะในการดำเนินการ เนื่องจากเป็นความร่วมมือจากทางสมาคมผู้ผลิตไก่เพื่อส่งออกไทยให้ส่งวัตถุดิบให้กับรงกำจัดซากไก่ โดยทำการแบ่งตามสัดส่วนของการผลิตในแต่ละจุด โดยตำแหน่งที่ตั้งของโรงกำจัดซากไก่ ฟาร์ม และโรงเชือด จะคิดจากพิกัดในระบบ GPS (หมายเลข EST. ย่อมาจาก Establishment Approved Number ในแต่ละจุดของลูกค้า แสดงถึง ลำดับหมายเลขโรงงานที่ได้รับการอนุมัติให้สามารถส่งออกต่างประเทศได้) โดยมีรายละเอียด ดังแสดงในตารางที่ 1-1 (ข้อมูลจากสำนักตรวจสอบฯ กรมปศุสัตว์)

3.1.2 การวัดระยะทางการขนส่ง

การวัดระยะทางการขนส่งจะวัดระยะทางที่แสดงตามแผนที่จาก Google (Google Map) และแสดงระยะทางระหว่างแต่ละจุด และระยะทางระหว่างโรงกำจัดซากทั้ง 2 ตำแหน่ง แล้วทำการบันทึกระยะทางลงในตารางเมตริกระยะทาง (Distance Matrix) ดังแสดงในตารางที่ 3-1 และ 3-2

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 3-1 ระยะทางระหว่างแต่ละจุด กรณีที่ตั้งโรงกำจัดซากไก่ที่ ต.ลำพญากลาง จ.ลพบุรี

ระยะทางระหว่างจุด (กม.)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	0	226	257	276	322	316	271	244	130	120	75.6	188	189	191	123	482	250	247	109	84.9	104	64.9	255
1	226	0	67.1	128	115	103	73.4	64.6	273	274	229	47.4	24.9	83.7	282	184	40.4	67.6	121	146	134	170	137
2	257	67.1	0	120	151	136	132	37.2	319	309	259	123	77.8	82.6	318	213	58.3	36.6	151	176	170	199	128
3	276	128	120	0	236	224	212	87.2	331	321	277	156	124	85.4	330	231	152	86.9	168	193	182	217	16.6
4	322	115	151	236	0	59.7	41.7	163	276	258	284	127	136	199	392	293	99.2	162	222	247	244	270	250
5	316	103	136	224	59.7	0	59	139	296	278	304	120	126	182	373	274	79.4	138	211	237	225	260	231
6	271	73.4	132	212	41.7	59	0	134	253	235	261	89.1	114	170	327	228	73.7	137	194	191	179	214	219
7	244	64.6	37.2	87.2	163	139	134	0	307	296	252	119	79	75.2	305	206	68.3	3	143	169	157	192	96
8	130	273	319	331	276	296	253	307	0	12.2	59.2	222	233	258	238	156	269	291	153	194	182	150	323
9	120	274	309	321	258	278	235	296	12.2	0	61.1	223	235	260	240	158	295	293	155	195	184	152	330
10	75.6	229	259	277	284	304	261	252	59.2	61.1	0	181	192	218	198	116	253	250	113	153	141	110	282
11	188	47.4	123	156	127	120	89.1	119	222	223	181	0	47.7	102	244	145	82.4	117	76.9	108	96.2	120	166
12	189	24.9	77.8	124	136	126	114	79	233	235	192	47.7	0	69.7	245	146	58	76.5	83.3	109	96.8	132	134

13	191	83.7	82.6	85.4	199	182	170	75.2	258	260	218	102	69.7	0	247	148	106	72.6	107	110	88.2	156	72.8
14	123	282	318	330	392	373	327	305	238	240	198	244	245	247	0	109	305	302	178	140	159	140	320
15	48.2	184	213	231	293	274	228	206	156	158	116	145	146	148	109	0	215	203	79.4	40.9	60	44.3	212
16	250	40.4	58.3	152	99.2	79.4	73.7	68.3	269	295	253	82.4	58	106	305	215	0	74.3	146	171	159	194	165
17	247	67.6	36.6	86.9	162	138	137	3	291	293	250	117	76.5	72.6	302	203	74.3	0	139	164	152	187	95.4
18	109	121	151	168	222	211	194	143	153	155	113	76.9	83.3	107	178	79.4	146	139	0	46.6	34	42.6	189
19	84.9	146	176	193	247	237	191	169	194	195	153	108	109	110	140	40.9	171	164	46.6	0	25.4	34.4	177
20	104	134	170	182	244	225	179	157	182	184	141	96.2	96.8	88.2	159	60	159	152	34	25.4	0	44	161
21	64.9	170	199	217	270	260	214	192	150	152	110	120	132	156	140	44.3	194	187	42.6	34.4	44	0	218
22	255	137	128	16.6	250	231	219	96	323	330	282	166	134	728	320	212	165	95.4	189	177	161	218	0

ตารางที่ 3-2 ระยะทางระหว่างแต่ละจุด กรณีที่ตั้งโรงกำจัดซากไก่ที่ ต.ทับกวาง จ.สระบุรี

ระยะทางระหว่างจุด (กม.)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	0	131	166	181	233	221	209	149	162	152	107	82.8	94.1	119	191	92.1	155	152	145	54.8	43	30.9	190
1	131	0	67.1	128	115	103	73.4	64.6	273	274	229	47.4	24.9	83.7	282	184	40.4	67.6	121	146	134	170	137
2	166	67.1	0	120	151	136	132	37.2	319	309	259	123	77.8	82.6	318	213	58.3	36.6	151	176	170	199	128

3	181	128	120	0	236	224	212	87.2	331	321	277	156	124	85.4	330	231	152	86.9	168	193	182	217	16.6
4	233	115	151	236	0	59.7	41.7	163	276	258	284	127	136	199	392	293	99.2	162	222	247	244	270	250
5	221	103	136	224	59.7	0	59	139	296	278	304	120	126	182	373	274	79.4	138	211	237	225	260	231
6	209	73.4	132	212	41.7	59	0	134	253	235	261	89.1	114	170	327	228	73.7	137	194	191	179	214	219
7	149	64.6	37.2	87.2	163	139	134	0	307	296	252	119	79	75.2	305	206	68.3	3	143	169	157	192	96
8	162	273	319	331	276	296	253	307	0	12.2	59.2	222	233	258	238	156	269	291	153	194	182	150	323
9	152	274	309	321	258	278	235	296	12.2	0	61.1	223	235	260	240	158	295	293	155	195	184	152	330
10	107	229	259	277	284	304	261	252	59.2	61.1	0	181	192	218	198	116	253	250	113	153	141	110	282
11	82.8	47.4	123	156	127	120	89.1	119	222	223	181	0	47.7	102	244	145	82.4	117	76.9	108	96.2	120	166
12	94.1	24.9	77.8	124	136	126	114	79	233	235	192	47.7	0	69.7	245	146	58	76.5	83.3	109	96.8	132	134
13	119	83.7	82.6	85.4	199	182	170	75.2	258	260	218	102	69.7	0	247	148	106	72.6	107	110	88.2	156	72.8
14	191	282	318	330	392	373	327	305	238	240	198	244	245	247	0	109	305	302	178	140	159	140	320
15	92.1	184	213	231	293	274	228	206	156	158	116	145	146	148	109	0	215	203	79.4	40.9	60	44.3	212
16	155	40.4	58.3	152	99.2	79.4	73.7	68.3	269	295	253	82.4	58	106	305	215	0	74.3	146	171	159	194	165

17	152	67.6	36.6	86.9	162	138	137	3	291	293	250	117	76.5	72.6	302	203	74.3	0	139	164	152	187	95.4
18	14.5	121	151	168	222	211	194	143	153	155	113	76.9	83.3	107	178	79.4	146	139	0	46.6	34	42.6	189
19	54.8	146	176	193	247	237	191	169	194	195	153	108	109	110	140	40.9	171	164	46.6	0	25.4	34.4	177
20	43	134	170	182	244	225	179	157	182	184	141	96.2	96.8	88.2	159	60	159	152	34	25.4	0	44	161
21	30.9	170	199	217	270	260	214	192	150	152	110	120	132	156	140	44.3	194	187	42.6	34.4	44	0	218
22	190	137	128	16.6	250	231	219	96	323	330	282	166	134	72.8	320	212	165	95.4	189	177	161	218	0

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.1.3 ปริมาณสินค้าประเภทซากไก่

ปริมาณสินค้าประเภทซากไก่ที่จะส่งเข้าโรงกำจัดซากไก่ในแต่ละวัน เป็นซากไก่ตายที่มาจากฟาร์ม ในแต่ละจุด คิดเป็นร้อยละ 3 ของปริมาณไก่ที่จะส่งเข้าโรงเชือดในแต่ละวัน และคิดน้ำหนักเฉลี่ยของไก่ตายตัวละ 1.3 กิโลกรัม (ข้อมูลจากสมาคมผู้ผลิตไก่เพื่อส่งออกไทย) ซึ่งเมื่อคำนวณปริมาณทั้งหมดแล้วพบว่าจะมีปริมาณเกินกำลังการผลิตของสายการผลิตที่ 1 (30,000 ตันต่อปี) จึงปรับปริมาณวัตถุดิบที่จะไปรับแต่ละจุดตามอัตราส่วน (proportional) ได้ปริมาณแต่ละจุด ดังแสดงในตารางที่ 3-3

3.1.4 ปริมาณสินค้าประเภทเศษซากไก่

ปริมาณสินค้าประเภทเศษซากไก่ที่จะส่งเข้าโรงกำจัดซากไก่ในแต่ละวัน เป็นเศษซากไก่ (โครงไก่) ที่มาจากโรงเชือด ในแต่ละจุด คิดโครงไก่เป็นร้อยละ 21 ของน้ำหนักตัวไก่ที่มีน้ำหนักเฉลี่ยตัวละ 2.5 กิโลกรัม และโรงเชือดตกลงที่จะส่งโครงไก่ร้อยละ 6 เป็นวัตถุดิบให้โรงกำจัดซากไก่ (ข้อมูลจากสมาคมผู้ผลิตไก่เพื่อส่งออกไทย) ซึ่งเมื่อคำนวณปริมาณทั้งหมดแล้วพบว่าจะมีปริมาณเกินกำลังการผลิตของสายการผลิตที่ 2 (30,000 ตันต่อปี) จึงปรับปริมาณวัตถุดิบที่จะไปรับแต่ละจุดตามอัตราส่วน (proportional) ได้ปริมาณแต่ละจุด ดังแสดงในตารางที่ 3-3

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 3-3 ปริมาณซากไก่และเศษซากไก่ (โครงไก่) ในแต่ละจุดที่จะส่งเป็นวัตถุดิบให้โรงกำจัดซากไก่ที่ปรับปริมาณวัตถุดิบที่จะไปรับแต่ละจุดตามอัตราส่วน

			[1]	[2]	[3]	[4]	[5]		[6]	[7]	[8]	[9]
จุดลูกค้า	**EST**	**จังหวัด**	**ปริมาณไก่ที่เข้าโรงเชือด (ตัว/วัน)**	**ปริมาณไก่ที่เลี้ยง (ตัว/วัน)**	**ปริมาณไก่ตาย (ตัว/วัน)**	**น้ำหนักไก่ตายที่ส่งโรงกำจัด (ตัน /วัน)**	**น้ำหนักไก่ตายที่ไปรับ (ตัน/วัน)**		**น้ำหนักไก่ที่เข้าโรงเชือด (ตัน/วัน)**	**น้ำหนักโครงไก่ที่ออกจากโรงเชือด (ตัน/วัน)**	**น้ำหนักโครงไก่ที่ส่งโรงกำจัด (ตัน/วัน)**	**น้ำหนักโครงไก่ที่ไปรับ (ตัน/วัน)**
1	3	กรุงเทพ	180,000	185,567	5,567	7.24	5.22		450	94.5	5.67	5.22
2	18	กรุงเทพ	60,000	61,856	1,856	2.41	1.74		150	31.5	1.89	1.74
3	119	กาญจนบุรี	340,000	350,515	10,515	13.67	9.86		850	178.5	10.71	9.86
4	201	ชลบุรี	100,000	103,093	3,093	4.02	2.90		250	52.5	3.15	2.90
5	109	ชลบุรี	110,000	113,402	3,402	4.42	3.19		275	57.75	3.465	3.19
6	79	ชลบุรี	130,000	134,021	4,021	5.23	3.77		325	68.25	4.095	3.77
7	6	นครปฐม	70,000	72,165	2,165	2.81	2.03		175	36.75	2.205	2.03
8	139	นครราชสีมา	260,000	268,041	8,041	10.45	7.54		650	136.5	8.19	7.54
9	165	นครราชสีมา	110,000	113,402	3,402	4.42	3.19		275	57.75	3.465	3.19
10	70	นครราชสีมา	130,000	134,021	4,021	5.23	3.77		325	68.25	4.095	3.77
11	53	นครนายก	90,000	92,784	2,784	3.62	2.61		225	47.25	2.835	2.61
12	34	ปทุมธานี	90,000	92,784	2,784	3.62	2.61		225	47.25	2.835	2.61
13	7	ปทุมธานี	90,000	92,784	2,784	3.62	2.61		225	47.25	2.835	2.61
14	129	เพชรบูรณ์	250,000	257,732	7,732	10.05	7.25		625	131.25	7.875	7.25
15	44	ลพบุรี	310,000	319,588	9,588	12.46	8.99		775	162.75	9.765	8.99
16	11	สมุทรปราการ	130,000	134,021	4,021	5.23	3.77		325	68.25	4.095	3.77
17	10	สมุทรสาคร	130,000	134,021	4,021	5.23	3.77		325	68.25	4.095	3.77
18	23	สระบุรี	240,000	247,423	7,423	9.65	6.96		600	126	7.56	6.96
19	49	สระบุรี	270,000	278,351	8,351	10.86	7.83		675	141.75	8.505	7.83
20	32	สระบุรี	220,000	226,804	6,804	8.85	6.38		550	115.5	6.93	6.38
21	25	สระบุรี	90,000	92,784	2,784	3.62	2.61		225	47.25	2.835	2.61
22	160	สุพรรณบุรี	50,000	51,546	1,546	2.01	1.45		125	26.25	1.575	1.45
รวม			**3,450,000**	**3,556,705**	**106,705**	138.72	100		8,625	1,811.25	108.68	100

จากตารางที่ 3-3 พบว่า จากการบันทึกปริมาณไก่ที่เข้าโรงเชือดของจุดลูกค้าที่ 1 ถึง 22 ซึ่งแสดงได้ดังช่องที่ 1 โดยจะแบ่งปริมาณไก่ที่เลี้ยงออกเป็น 2 ส่วน คือ ปริมาณไก่ตาย และ ปริมาณไก่ที่เข้าโรงเชือด ซึ่งมีสัดส่วนเป็น ร้อยละ 3 และ 97 ของปริมาณไก่ที่เลี้ยงในแต่ละวัน ตามลำดับ ซึ่งทำให้ได้ปริมาณไก่ตายเฉลี่ยต่อวันดังช่องที่ 3 และทำให้ได้ปริมาณไก่ที่เข้าโรงเชือดดังช่องที่ 1 และจากข้อมูลสมาคมผู้ผลิตไก่เพื่อส่งออกไทยระบุว่า น้ำหนักเฉลี่ยของไก่ตายตัวละ 1.3 กิโลกรัม และน้ำหนักไก่ที่ก่อนเข้าโรงเชือดเฉลี่ยตัวละ 2.5 กิโลกรัม

ช่องที่ [2] = ช่องที่ [1] x (1/0.97)
ช่องที่ [3] = ช่องที่ [2] x 0.03

ทำให้สามารถแสดงปริมาณน้ำหนักไก่ตายที่ส่งโรงกำจัดได้ดังช่องที่ 4 และแสดงน้ำหนักไก่ที่เข้าโรงเชือดดังช่องที่ 6

ช่องที่ [4] = ช่องที่ [3] x (1.3/1000)
ช่องที่ [6] = ช่องที่ [2] x 0.97 x (2.5/1000)

จากข้อมูลสมาคมผู้ผลิตไก่เพื่อส่งออกไทยระบุว่า น้ำหนักของโครงไก่คิดเป็นร้อยละ 21 ของน้ำหนักตัวไก่ ทำให้ได้ปริมาณน้ำหนักโครงไก่ที่ออกจากโรงเชือดดังช่องที่ 7 และ โรงกำจัดซากไก่สามารถรับโครงไก่ได้เพียงร้อยละ 6 ของปริมาณน้ำหนักโครงไก่ที่ออกจากโรงเชือด ซึ่งแสดงดังช่องที่ 8

ช่องที่ [7] = ช่องที่ [6] x 0.21
ช่องที่ [8] = ช่องที่ [7] x 0.06

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

และจากผลรวมของน้ำหนักไก่ตายที่ส่งโรงกำจัดซากไก่ 138.72 ตัน/วัน และผลรวมของน้ำหนักโครงไก่ที่ส่งโรงกำจัดซากไก่ 108.68 ตัน/วัน จะเห็นได้ว่า เกินกำลังการผลิตของโรงกำจัดซากไกในแต่ละสายการผลิตที่ 100 ตัน/วัน จึงต้องทำการจัดสรรสัดส่วนปริมาณน้ำหนักไก่ตายที่ไปรับและน้ำหนักโครงไก่ที่ไปรับในแต่ละจุดดังช่องที่ 5 และ 9

ช่องที่ [5] = ช่องที่ [4] x (100/ผลรวมช่องที่ [4])
ช่องที่ [9] = ช่องที่ [8] x (100/ผลรวมช่องที่ [8])

3.1.5 รถที่ใช้ในการขนส่ง

รถที่ใช้ในการขนส่งจะเป็นรถตู้เย็นมีขนาดบรรทุกประมาณ 12 ตัน โดยจะแยกรถที่ใช้ขนวัตถุดิบประเภทซากไก่ตาย และรถที่ใช้ขนเศษซากไก่ (โครงไก่) จากโรงเชือด โดยไม่ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ และรถขนส่งที่ใช้จะเป็นรถรับจ้างแบบเหมาจ่ายแบบขั้นบันได (Stepwise Linear) ตามระยะทางที่ขนส่ง (ข้อมูลจากสมาคมผู้ผลิตไก่เพื่อส่งออกไทย) ดังแสดงในตารางที่ 3-4

ตารางที่ 3-4 อัตราค่าขนส่งของรถบรรทุก 10 ล้อ แบบห้องเย็นขนาด 12 ตัน

ระยะทางรวม (กิโลเมตร)	**ราคาเหมาจ่ายต่อเที่ยว (บาท)**
100 - 0	3,740
200 - 100	5,270
300 - 200	6,700
400 - 300	8,300
500 - 400	9,800
600 - 500	11,300
700 - 600	13,000
800 - 700	14,400
900 - 800	16,000
1,000 – 900	17,500

3.1.6 การคำนวณเวลาในการเดินทาง

การคำนวณเวลาในการเดินทางจากแต่ละจุดจะคิดที่ความเร็วเฉลี่ย 60 กิโลเมตรต่อชั่วโมง

3.1.7 การคำนวณเวลาในการขนสินค้า

การคำนวณเวลาในการขนสินค้าขึ้นลงจากรถบรรทุกในแต่ละจุดจะคิดเป็นสัดส่วนกับปริมาณสินค้าที่จะขนในแต่ละจุด ใช้เวลาขนขึ้นลงประมาณ 20 นาทีต่อตัน และกำหนดกรอบเวลาในการขนส่งทั้งหมดให้เสร็จสิ้นภายใน 24 ชั่วโมง

3.1.8 การคิดค่าใช้จ่ายในการขนส่งต่อปี

การคิดค่าใช้จ่ายในการขนส่งต่อปีจะคิดจากวันทำงานเฉลี่ยปีละ 300 วัน

3.1.9 การวิ่งของรถที่วิ่งไปรับสินค้าในแต่ละรอบการวิ่ง

การวิ่งของรถที่วิ่งไปรับสินค้าในแต่ละรอบการวิ่ง จะคิดระยะทางตั้งต้นที่โรงกำจัดซากไก่ (ในการคำนวณจะเรียกว่า Depot) ไปยังจุดรับสินค้าแต่ละจุด (หากรับหลายจุด จะรับจนครบตามเส้นทางที่กำหนด) และวิ่งกลับไปส่งสินค้าที่ Depot

3.1.10 การจัดรอบการวิ่งและลำดับเส้นทาง

การจัดรอบการวิ่งและลำดับเส้นทาง คิดจากสมมุติฐานที่มีการเดินรถเหมือนกันทุกวัน หากปริมาณที่ขนส่งบางจุดในแต่ละวันไม่เต็มความจุของรถบรรทุก อาจมีบางรอบที่รถบรรทุกต้องไปรับวัตถุดิบจากหลายจุด (multiple pickup locations)

3.2 แบบจำลองทางคณิตศาสตร์ (Mathematical model)

การกำหนดดัชนี

$i = \{0,1,2,\ldots,N\}$ เป็นเซตของลูกค้า i
$j = \{0,1,2,\ldots,N\}$ เป็นเซตของลูกค้า j
$k = \{0,1,2,\ldots,N\}$ เป็นเซตของลูกค้า k
$c = \{1,2,\ldots,C\}$ เป็นเซตของรถบรรทุกทั้งหมด c
$r = \{1,2,\ldots,R\}$ เป็นเซตของรอบการวิ่งทั้งหมด r
$e = \{1,2,\ldots,E\}$ เป็นเซตของช่วงของราคารถ e

การกำหนดพารามิเตอร์

$dist_{ij}$ = ระยะทางจากลูกค้า i ไปลูกค้า j (กิโลเมตร)
t_{ij} = ระยะเวลาการเดินทางจากลูกค้า i ไปลูกค้า j (นาที)
w_i = น้ำหนักของสินค้าทั้งหมดที่ต้องรับจากลูกค้า i (ตัน/วัน)
Lt_j = ระยะเวลาขนสินค้าขึ้นรถ ณ ลูกค้า j (นาที/ตัน)
Cap_c = ความสามารถในการบรรทุกของรถ c (ตัน/รอบการวิ่ง)
$LimitTime$ = กรอบเวลาในการขนส่งสูงสุด (นาที)
M = Big M Parameter
$distedge_e$ = ช่วงระยะการเดินทางของรถมากที่สุดในช่วงระยะเหมาจ่ายแบบขั้นบันได e (กิโลเมตร)
$costedge_e$ = ค่าใช้จ่ายของระยะการเดินทางของรถในช่วงระยะเหมาจ่ายแบบขั้นบันได e (บาท)

ตัวแปรตัดสินใจ

$$X_{ij}^{cr} = \begin{cases} 1 & \text{กรณีที่มีการขนส่งโดยรถคันที่ } c \text{ รอบการวิ่งที่ } r \text{ โดยเดินทางจากลูกค้า } i \text{ ไป } j \\ 0 & \text{กรณีที่ไม่เกิดการขนส่งขึ้น} \end{cases}$$

beg_{cr} = เวลาออกเดินทางจากโรงกำจัดซากไก่ของรถ คันที่ c รอบการวิ่งที่ r
end_{cr} = เวลาที่กลับมาถึงโรงกำจัดซากไก่ของรถคันที่ c รอบการวิ่งที่ r

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

WV_{icr} = น้ำหนักของสินค้าจากลูกค้า i ที่บรรทุก โดย คันที่ c รอบการวิ่งที่ r

CWV_{ijcr} = น้ำหนักที่สะสมของสินค้าจากลูกค้า i ไปลูกค้า j โดยรถคันที่ c รอบการวิ่งที่ r

u_{icr} = การป้องกันการเดินรถย่อย (Sub-Tour) $\in Z^+$

$$zedge_{cre} = \begin{cases} 1 & \text{กรณีที่รถคันที่ } c \text{ รอบการวิ่งที่ } r \text{ วิ่งด้วยระยะทางภายในช่วงของราคารถ } e \\ 0 & \text{กรณีที่รถคันที่ } c \text{ รอบการวิ่งที่ } r \text{ ไม่ได้วิ่งด้วยระยะทางภายในช่วงของราคารถ } e \end{cases}$$

ซึ่งเขียนแบบจำลองทางคณิตศาสตร์ ได้ดังนี้

ความสัมพันธ์ (12) เป็นฟังก์ชั่นวัตถุประสงค์ โดยมีวัตถุประสงค์ในการขนส่งเป็นอัตราเหมาจ่ายแบบขั้นบันไดตามระยะทางที่มีต้นทุนต่ำที่สุด ดังนี้

$$Minimize \sum_c \sum_r \sum_e \cos tedge_e * zedge_{cre} \quad (12)$$

ความสัมพันธ์ (13) เป็นเงื่อนไข แสดงน้ำหนักรวมที่รถทุกคันทุกรอบไปรับจากจุด i ต้องไม่น้อยกว่าน้ำหนักที่กำหนดของจุด i ดังนี้

$$\sum_r \sum_c WV_{icr} \geq W_i \qquad \forall_i \quad (13)$$

ความสัมพันธ์ (14) และ (15) เป็นเงื่อนไขบังคับให้รถวิ่งออกจากโรงกำจัดซากไก่ ไปยังลูกค้าจุดต่างๆแล้วกลับมาที่โรงกำจัดซากไก่เสมอ ไม่ให้มีการวิ่งวนโดยไม่ผ่านโรงกำจัดซากไก่ (ป้องกันการ Sub-Tour) ดังนี้

$$u_{icr} + u_{jcr} + n * X_{ij}^{cr} \leq n-1 \quad \forall_i \forall_j \forall_c \forall_r ; \; i \neq 0, j \neq 0 \quad *n \text{ is No.nodes.} \quad (14)$$

$$\sum_{j=0}^{n} X_{ij}^{cr} - \sum_{k=0}^{n} X_{ki}^{cr} = 0 \quad \forall_i \; \forall_c \; \forall_r \quad (15)$$

ความสัมพันธ์ (16) เป็นเงื่อนไขบังคับให้น้ำหนักที่บรรทุกออกจากที่จุดใดๆ จะเท่ากับน้ำหนักที่รถวิ่งเข้าจุดนั้นรวมกับน้ำหนักที่บรรทุกจากจุดนั้นเสมอ โดยมี Big M Parameter ใช้สำหรับบังคับในกรณีที่รถวิ่งออกจากจุดเริ่มต้นไปที่จุดเดียวแล้วกลับมาที่จุดเริ่มต้น ดังนี้

$$\sum_i CWV_{ijcr} + WV_{jcr} \leq \sum_k CWV_{jkcr} + M * X_{j0}^{cr} \quad \forall_j \forall_c \forall_r ; \; i \neq j \neq k \quad (16)$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ความสัมพันธ์ (17) เป็นเงื่อนไขที่กำหนดให้น้ำหนักที่รถบรรทุกในแต่ละรอบการวิ่งรับสินค้าไม่เกินขีดจำกัดในการบรรทุกของรถ ดังนี้

$$CWV_{ijcr} \leq Cap_c \quad \forall_i \forall_j \forall_c \forall_r \qquad (17)$$

ความสัมพันธ์ (18) เป็นเงื่อนไขที่น้ำหนักบรรทุก ณ จุดใดๆที่รับขึ้นรถไม่เกินความสามารถในการบรรทุกของรถ ดังนี้

$$WV_{jcr} \leq Cap_c * \sum_i X_{ij}^{cr} \quad \forall_j \forall_c \forall_r \qquad (18)$$

ความสัมพันธ์ (19) เป็นเงื่อนไขที่กำหนดให้เวลาสิ้นสุดการวิ่งในแต่ละรอบเท่ากับเวลาเริ่มการวิ่งจากโรงกำจัดซากไก่รวมกับเวลาในการวิ่งระหว่างจุดภายในรอบการวิ่งและเวลาที่ใช้ในการเคลื่อนย้ายสินค้าของที่จุดนั้นๆ ดังนี้

$$\sum_i \sum_j t_{ij} * X_{ij}^{cr} + \sum_j Lt_j * WV_{jcr} + beg_{cr} = end_{cr} \quad \forall_c \forall_r \qquad (19)$$

ความสัมพันธ์ (20) เป็นเงื่อนไขที่กำหนดให้เวลาเริ่มการวิ่งในรอบถัดไปของรถแต่ละคันมีค่าไม่ต่ำกว่าเวลาสิ้นสุดการวิ่งของรอบการวิ่งก่อนหน้า ดังนี้

$$beg_{c,r+1} \geq end_{cr} \quad \forall_c \forall_r = 1,\ldots,number_{r-1} \qquad (20)$$

ความสัมพันธ์ (21) เป็นเงื่อนไขที่กำหนดไม่ให้มีการวิ่งข้ามรอบ เช่น วิ่งรอบที่ 2 ก่อนรอบที่ 1 เป็นต้น ดังนี้

$$\sum_j X_{0j}^{cr} \geq \sum_j X_{0j}^{c,r+1} \quad \forall_c \forall_r = 1,\ldots,number_{r-1} \qquad (21)$$

ความสัมพันธ์ (22) เป็นเงื่อนไขที่กำหนดให้เวลาสิ้นสุดการวิ่งของรถต้องมีค่าไม่เกินข้อกำหนดของเวลาที่อนุญาตให้วิ่งรถได้ ดังนี้

$$end_{cr} \leq LimitTime \quad \forall_c \forall_r \qquad (22)$$

ความสัมพันธ์ (23) และ(24) เป็นเงื่อนไขที่กำหนดให้ช่วงราคาแบบเหมาจ่ายของระยะทางในการขนส่งในแต่ละรอบการวิ่ง โดยการวิ่งในแต่ละรอบจะอยู่ในช่วงราคาช่วงใดช่วงหนึ่งเท่านั้น ดังนี้

$$\sum_i \sum_j dist_{ij} * X_{ij}^{cr} \leq \sum_e distedge_e * zedge_{cre} \quad \forall_c \forall_r \qquad (23)$$

$$\sum_e zedge_{cre} \leq 1 \quad \forall_c \forall_r \qquad (24)$$

ในงานวิจัยนี้จะทำการแก้ปัญหาจากแบบจำลองทางคณิตศาสตร์ที่ได้ทำการออกแบบด้วยการใช้โปรแกรม IBM ILOG CPLEX OPTIMIZATION STUDIO (รหัสคำสั่งโปรแกรมแสดงในภาคผนวก ข) โดยผลลัพธ์จากการจำลอง ดังแสดงในตารางที่ 3-5 ถึง 3-10

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 3-5 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 5 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot --> ซีพีเอฟ (est.18) [1.74] --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	11.6	653	885	13,000
2	Depot --> ซีพีเอฟ (est.03) [5.22] --> GFN (est.201) [2.90] --> ไทยโพลทรี กรุ๊ป (est.109) [3.19] --> Depot	11.31	716.7	942.9	14,400
	รวม	22.91	1,369.7	1,827.9	27,400

จากตารางที่ 3-5 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ลำพญากลาง ไปรับซากไก่และเศษซากไก่จาก 5 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบที่ดีที่สุด คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 54,800 บาทต่อวัน และใช้ระยะทางทั้งหมด 2,739.4 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 1.23 นาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-6 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 4 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot --> GFN (est.201) [2.90] --> ซีพีเอฟ (est.03) [5.22] --> ซีพีเอฟ (est.18) [1.74] --> Depot	9.86	761.1	958.3	14,400
2	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	9.86	552	749.2	11,300
	รวม	19.72	1,313.1	1,707.5	25,700

จากตารางที่ 3-6 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ลำพญากลาง ไปรับซากไก่และเศษซากไก่จาก 4 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบที่ดีที่สุด คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 51,400 บาทต่อวัน และใช้ระยะทางทั้งหมด 2,626.2 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 39 วินาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-7 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 3 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot --> ซีพีเอฟ (est.03) [5.22] --> ซีพีเอฟ (est.18) [1.74] --> Depot	6.96	550.1	689.3	11,300
2	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	9.86	552	749.2	11,300
	รวม	16.82	1,102.1	1,438.5	22,600

จากตารางที่ 3-7 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ลำพญากลาง ไปรับซากไก่และเศษซากไก่จาก 3 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบที่ดีที่สุด คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 45,200 บาทต่อวัน และใช้ระยะทางทั้งหมด 2,204.2 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 9 วินาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-8 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 5 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> ซีพีเอฟ (est.18) [1.74] --> Depot	11.6	467	699	9,800
2	Depot --> ซีพีเอฟ (est.03) [5.22] --> ไทยโพลทรีกรุ๊ป (est.109) [3.19] --> GFN (est.201) [2.90] --> Depot	11.31	526.7	752.9	11,300
	รวม	22.91	993.7	1,451.9	21,100

จากตารางที่ 3-8 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ทับกวาง ไปรับซากไก่และเศษซากไก่จาก 5 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบที่ดีที่สุด คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 42,200 บาทต่อวัน และใช้ระยะทางทั้งหมด 1,987.4 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 1.57 นาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-9 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 4 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot --> ซีพีเอฟ (est.18) [1.74] --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	11.6	467	699	9,800
2	Depot --> GFN (est.201) [2.90] --> ซีพีเอฟ (est.03) [5.22] -->Depot	8.12	479	641.4	9,800
	รวม	19.72	946	1,340.4	19,600

จากตารางที่ 3-9 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ทับกวาง ไปรับซากไก่และเศษซากไก่จาก 4 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบที่ดีที่สุด คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 39,200 บาทต่อวัน และใช้ระยะทางทั้งหมด 1,892 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 1 นาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-10 ผลลัพธ์จากแบบจำลองโดยรับซากและเศษซากไก่ทั้งหมด 3 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot --> ซีพีเอฟ (est.18) [1.74] --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	11.6	467	699	9,800
2	Depot --> ซีพีเอฟ (est.03) [5.22] --> Depot	5.22	262	366.4	6,700
	รวม	16.82	729	1,065.4	16,500

จากตารางที่ 3-10 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ทับกวาง ไปรับซากไก่และเศษซากไก่จาก 3 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบที่ดีที่สุด คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 33,000 บาทต่อวัน และใช้ระยะทางทั้งหมด 1,458 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 13 วินาที เหมือนกันทั้ง 2 กรณี

จากนั้นเมื่อขยายปัญหาให้มีจำนวนจุดที่มากขึ้น พบว่า เวลาที่ใช้ในการคำนวณเพิ่มขึ้นมาก จึงได้ทำการหาวิธีการพัฒนาการหาคำตอบที่ดีและรวดเร็วโดยการนำวิธีทางฮิวริสติกมาประยุกต์ในการหาคำตอบ โดยแบ่งออกเป็น 2 ระยะ คือ ระยะแรกเป็นการสร้างคำตอบตั้งต้นด้วย Saving Algorithm สำหรับในระยะที่สองเป็นการปรับปรุงคำตอบตั้งต้นที่ได้จากระยะแรกด้วย Genetic Algorithm ซึ่งจะกล่าวในบทที่ 4 ต่อไป

3.3 การสร้างคำตอบตั้งต้น

การสร้างคำตอบตั้งต้น (Initial solution) ที่ดีทำให้มีโอกาสได้ผลลัพธ์ที่ใกล้เคียงกับค่าคำตอบที่ดีที่สุด (Optimal solution) ดังนั้นผู้วิจัยจึงได้ประยุกต์ Saving Algorithm ในการสร้างเส้นทางตั้งต้น เนื่องจาก [14] มีการนำ Saving Algorithm มาใช้ในการจัดเส้นทางซึ่งให้ผลแตกต่างจากค่าคำตอบที่ดีที่สุดไม่เกิน 5% ทำให้เราเลือก Saving Algorithm เป็นคำตอบตั้งต้น

โดยขั้นตอนของ Saving Algorithm มีดังนี้

1. คำนวณระยะห่างของแต่ละจุดและทำการสร้างเมตริกระยะทาง (Distance Matrix) ระหว่างลูกค้าทั้งหมด ดังตารางที่ 3-2 และ 3-3
2. นำระยะทางของแต่ละคู่ของลูกค้ามาคำนวณ โดยการสร้างเมตริกแบบประหยัด (Saving Matrix) เพื่อหาค่าความประหยัดของแต่ละคู่ของลูกค้า ตามสมการที่ 10 ดังตารางที่ 3-11 และ 3-12

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตารางที่ 3-11 คำนวณค่าความประหยัดระหว่างแต่ละจุด กรณีที่ตั้งโรงกำจัดซากไก่ที่ ต.ลำพญากลาง จ.ลพบุรี

ค่าความประหยัด (กม.)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																
1		0														
2		416	0													
3		374	413	0												
4		433	428	362	0											
5		439	437	368	578	0										
6		424	396	335	551	528	0									
7		405	464	433	403	421	381	0								
8		83	68	75	176	150	148	67	0							
9		72	68	75	184	158	156	68	238	0						
10		72.6	73.6	75	114	87.6	85.6	67.6	146	135	0					
11		367	322	308	383	384	370	313	96	85	82.6	0				
12		390	368	341	375	379	346	354	86	74	72.6	329	0			
13		333	365	382	314	325	292	360	63	51	48.6	277	310	0		
14		67	62	69	53	66	67	62	15	3	0.6	67	67	67	0	
15		90.2	92.2	93	77.2	90.2	91.2	86.2	22.2	10.2	7.8	91.2	91.2	91.2	62	0
16		436	449	374	473	487	447	426	111	75	72.6	356	381	335	68	83.2
17		405	467	436	407	425	381	488	86	74	72.6	318	360	365	68	92.2
18		214	215	217	209	214	186	210	86	74	71.6	220	215	193	54	77.8
19		165	166	168	160	164	165	160	20.9	9.9	7.5	165	165	166	68	92.2
20		196	191	198	182	195	196	191	52	40	38.6	196	196	207	68	92.2
21		121	123	124	117	121	122	117	44.9	32.9	30.5	133	122	99.9	48	68.8
22		344	384	514	327	340	307	403	62	45	48.6	277	310	373	58	91.2

ตารางที่ 3-12 คำนวณค่าความประหยัดระหว่างแต่ละจุด กรณีที่ตั้งโรงกำจัดซากไก่ที่ ต.ทับกวาง จ.สระบุรี

ค่าความประหยัด (กม.)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	1
0																
1		0														
2		297	0													
3		184	227	0												
4		249	248	178	0											
5		249	251	178	394	0										
6		267	243	178	400	371	0									
7		215	278	243	219	231	224	0								
8		20	9	12	119	87	118	4	0							
9		9	9	12	127	95	126	5	302	0						
10		9	14	11	56	24	55	4	210	198	0					
11		166	126	108	189	184	203	113	22.8	11.8	8.8	0				
12		200	182	151	191	189	189	164	23.1	11.1	9.1	129	0			
13		166	202	215	153	158	158	193	23	11	8	99.8	143	0		

CHULALONGKORN UNIVERSITY

14		40	39	42	32	39	73	35	115	103	100	29.8	40.1	63	0	
15		39.1	45.1	42	32.1	39.1	73.1	35.1	98.1	86.1	83.1	29.9	40.2	63.1	174	
16		246	263	184	289	297	290	236	48	12	9	155	191	168	41	32
17		215	281	246	223	235	224	298	23	11	9	118	170	198	41	41
18		24.5	29.5	28	25.5	24.5	29.5	20.5	23.5	11.5	8.5	20.4	25.3	26.5	28	27
19		39.8	44.8	43	40.8	38.8	72.8	34.8	22.8	11.8	8.8	29.6	39.9	63.8	106	10
20		40	39	42	32	39	73	35	23	11	9	29.6	40.3	73.8	75	75
21		-8.1	-2.1	-5	-6.1	-8.1	25.9	-12	42.9	30.9	27.9	-6.3	-7	-6.1	82	78
22		184	228	354	173	180	180	243	29	12	15	107	150	236	61	70

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3. เริ่มทำการจัดเรียงเส้นทางโดยเรียงตามจากค่าความประหยัดที่ได้จากมากไปยังน้อยตามลำดับ

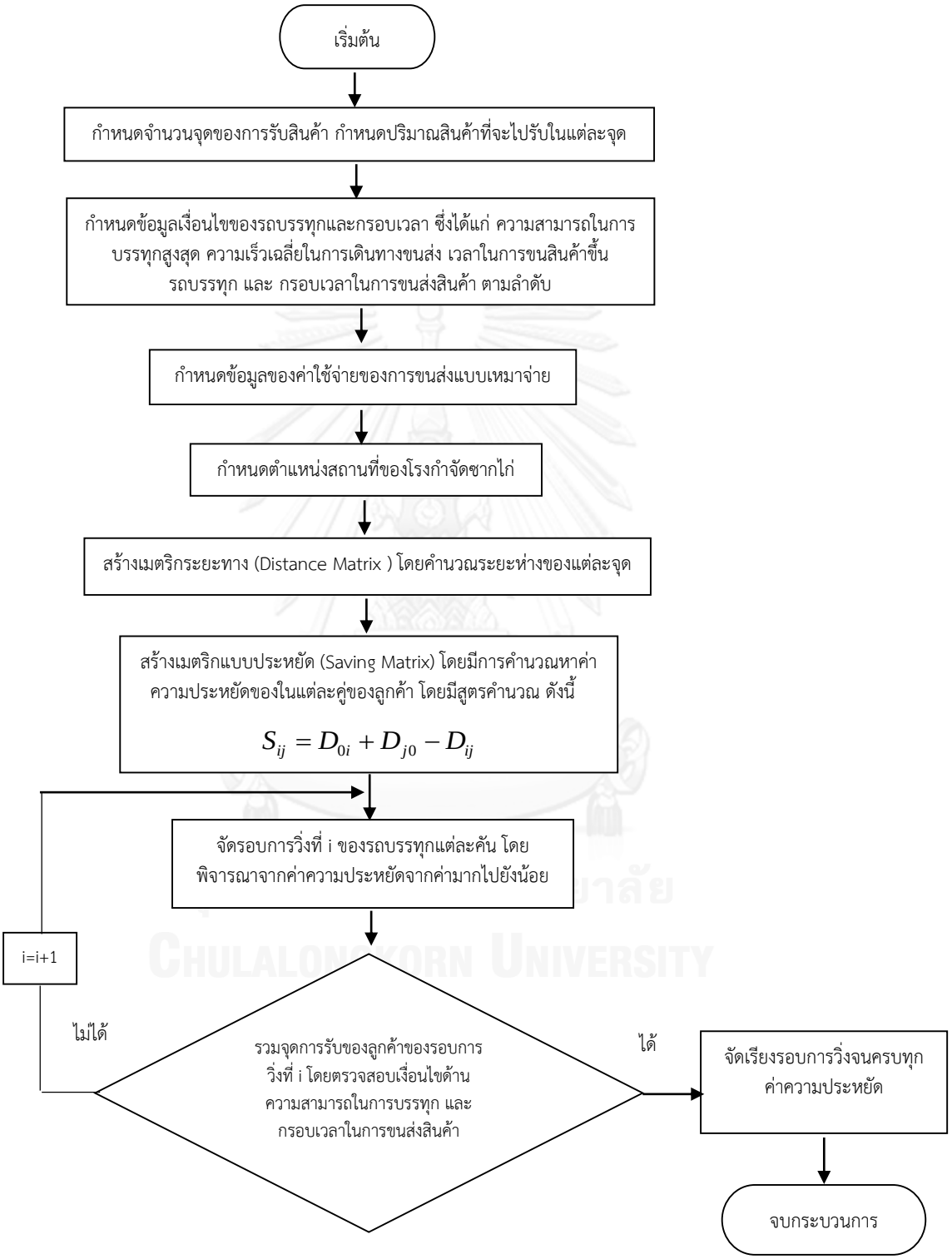
4. ถ้าจำนวนสินค้าทั้งหมดของทั้งสองเส้นทางไม่เกินความสามารถในการบรรทุกของรถและไม่เกินกรอบเวลาในการขนส่ง รอบการวิ่งทั้งสองจะถูกนำมารวมเป็นรอบการวิ่งเดียวกัน

5. ถ้าไม่สามารถเพิ่มจำนวนลูกค้าเข้ามาในเส้นทางได้อันเนื่องจาก เช่น ความสามารถในการบรรทุกของรถไม่เพียงพอ หรือ ข้อจำกัดด้านกรอบเวลาในการขนส่ง เป็นต้น จึงเริ่มจัดเส้นทางใหม่

6. ทำวนซ้ำไปเรื่อยๆจนครบทุกค่าความประหยัด จึงหยุดกระบวนการ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

โดยขั้นตอนการสร้างคำตอบตั้งต้นโดย Saving Algorithm สามารถแสดงเป็นแผนผัง ดังแสดงในรูปที่ 3-1



รูปที่ 3-1 แผนผังขั้นตอนการหาคำตอบตั้งต้นโดย Saving Algorithm

ในงานวิจัยนี้จะทำการแก้ปัญหาเพื่อหาคำตอบตั้งต้นโดยวิธี Saving Algorithm ที่ได้ทำการออกแบบด้วยการใช้ JAVA Script (รหัสคำสั่งโปรแกรมแสดงใน ภาคผนวก ค) โดยผลลัพธ์จากการประมวลผลในเบื้องต้นจะแสดงสำหรับการเดินทางไปรับสินค้าจำนวน 5 4 และ 3 จุด ตามลำดับ เพื่อเปรียบเทียบกับผลลัพธ์ของค่าที่ดีที่สุด ดังแสดงในตารางที่ 3-13 ถึง 3-18

ตารางที่ 3-13 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 5 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง)

รอบการวิ่ง	**เส้นทาง**	**น้ำหนัก (ตัน)**	**ระยะทาง (กม.ต่อวัน)**	**เวลา (นาที)**	**ค่าใช้จ่าย (บาทต่อวัน)**
1	Depot -->ไทยโพลทรีกรุ๊ป (est.109) [3.19] --> GFN (est.201) [2.90] --> Depot	6.09	697.7	819.5	13,000
2	Depot --> ซีพีเอฟ (est.03) [5.22] --> ซีพีเอฟ (est.18) [1.74] -->Depot	6.96	550.1	689.3	11,300
3	Depot -->ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	9.86	552	749.2	11,300
	รวม	22.91	1,799.8	2,258	35,600

จากตารางที่ 3-13 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ลำพญากลาง ไปรับซากไก่และเศษซากไก่จาก 5 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบจาก Saving Algorithm คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 71,200 บาทต่อวัน และใช้ระยะทางทั้งหมด 3,599.6 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 0.9 วินาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-14 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 4 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง)

รอบการวิ่ง	**เส้นทาง**	**น้ำหนัก (ตัน)**	**ระยะทาง (กม.ต่อวัน)**	**เวลา (นาที)**	**ค่าใช้จ่าย (บาทต่อวัน)**
1	Depot -->ซีพีเอฟ (est.03) [5.22] --> GFN (est.201) [2.90] --> Depot	8.12	663	825.4	13,000
2	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> ซีพีเอฟ (est.18) [1.74] --> Depot	11.60	653	885	13,000
	รวม	19.72	1,316	1,710.4	26,000

จากตารางที่ 3-14 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ลำพญากลาง ไปรับซากไก่และเศษซากไก่จาก 4 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบจาก Saving Algorithm คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 52,000 บาทต่อวัน และใช้ระยะทางทั้งหมด 2,632 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 0.6 วินาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-15 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 3 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot --> ซีพีเอฟ (est.03) [5.22] --> ซีพีเอฟ (est.18) [1.74] --> Depot	6.96	550.1	689.3	11,300
2	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	9.86	552	749.2	11,300
	รวม	16.82	1,102.1	1,438.5	22,600

จากตารางที่ 3-15 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ลำพญากลาง ไปรับซากไก่และเศษซากไก่จาก 3 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบจาก Saving Algorithm คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 45,200 บาทต่อวัน และใช้ระยะทางทั้งหมด 2,204.2 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 0.4 วินาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-16 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 5 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot -->ไทยโพลทรีกรุ๊ป (est.109) [3.19] --> GFN (est.201) [2.90] --> Depot	6.09	513.7	635.5	11,300
2	Depot --> ซีพีเอฟ (est.03) [5.22] --> ซีพีเอฟ (est.18) [1.74] --> Depot	6.96	364.1	503.3	8,300
3	Depot -->ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	9.86	362	559.2	8,300
	รวม	22.91	1,239.8	1,698	27,900

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตารางที่ 3-16 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ทับกวาง ไปรับซากไก่และเศษซากไก่จาก 5 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบจาก Saving Algorithm คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 55,800 บาทต่อวัน และใช้ระยะทางทั้งหมด 2,479.6 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 0.8 วินาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-17 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 4 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง)

รอบการวิ่ง	**เส้นทาง**	**น้ำหนัก (ตัน)**	**ระยะทาง (กม.ต่อวัน)**	**เวลา (นาที)**	**ค่าใช้จ่าย (บาทต่อวัน)**
1	Depot --> ซีพีเอฟ (est.03) [5.22] --> GFN (est.201) [2.90] --> Depot	8.12	479	641.4	9,800
2	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> ซีพีเอฟ (est.18) [1.74] --> Depot	11.60	467	699	9,800
	รวม	19.72	946	1,340.4	19,600

จากตารางที่ 3-17 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ทับกวาง ไปรับซากไก่และเศษซากไก่จาก 4 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบจาก Saving Algorithm คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 39,200 บาทต่อวัน และใช้ระยะทางทั้งหมด 1,892 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 0.6 วินาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-18 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 3 จุด (กรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง)

รอบการวิ่ง	**เส้นทาง**	**น้ำหนัก (ตัน)**	**ระยะทาง (กม.ต่อวัน)**	**เวลา (นาที)**	**ค่าใช้จ่าย (บาทต่อวัน)**
1	Depot --> ซีพีเอฟ (est.03) [5.22] --> ซีพีเอฟ (est.18) [1.74] --> Depot	6.96	364.1	503.3	8,300
2	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	9.86	362	559.2	8,300
	รวม	16.82	726.1	1,062.5	16,600

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จากตารางที่ 3-18 พบว่า การเดินทางจากโรงกำจัดซากไก่ที่ตั้งบริเวณ ต.ทับกวาง ไปรับซากไก่และเศษซากไก่จาก 3 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ ได้คำตอบจาก Saving Algorithm คือ ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมด 33,200 บาทต่อวัน และใช้ระยะทางทั้งหมด 1,452.2 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 0.3 วินาที เหมือนกันทั้ง 2 กรณี

จากนั้นจึงทำการเปรียบเทียบผลลัพธ์ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมดและระยะทางทั้งหมดระหว่างวิธีการหาค่าที่ดีที่สุดกับ Saving Algorithm ของการไปรับสินค้า 5 4 และ 3 จุด ดังแสดงในตารางที่ 3-19 ถึง 3-20

ตารางที่ 3-19 เปรียบเทียบผลลัพธ์ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมดและระยะทางทั้งหมดของการไปรับซากและเศษซากไก่ ระหว่างวิธีการหาค่าที่ดีที่สุดกับ Saving Algorithm (กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง)

จำนวนจุดที่รับ	วิธีการ	ค่าใช้จ่าย (บาทต่อวัน)	ความแตกต่างของค่าใช้จ่าย	ระยะทาง (กม.ต่อวัน)	ความแตกต่างของระยะทาง
5	วิธีหาค่าที่ดีที่สุด	54,800	23.03 %	2,739.4	23.9 %
	Saving Algorithm	71,200		3,599.6	
4	วิธีหาค่าที่ดีที่สุด	51,400	1.15 %	2,626.2	0.22 %
	Saving Algorithm	52,000		2,632	
3	วิธีหาค่าที่ดีที่สุด	45,200	0 %	2,204.2	0 %
	Saving Algorithm	45,200		2,204.2	

จากตารางที่ 3-19 พบว่า กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง มีความแตกต่างของผลลัพธ์ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมดและระยะทางทั้งหมดของการไปรับซากไก่ใน 5 4 และ 3 จุด คือ ประมาณร้อยละ 23 1 และ ไม่มีความแตกต่างกัน ตามลำดับ

ตารางที่ 3-20 เปรียบเทียบผลลัพธ์ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมดและระยะทางทั้งหมดของการไปรับซากและเศษซากไก่ ระหว่างวิธีการหาค่าที่ดีที่สุดกับ Saving Algorithm (กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง)

จำนวนจุดที่รับ	วิธีการ	ค่าใช้จ่าย (บาทต่อวัน)	ความแตกต่างของค่าใช้จ่าย	ระยะทาง (กม.ต่อวัน)	ความแตกต่างของระยะทาง
5	วิธีหาค่าที่ดีที่สุด	42,200	24.37 %	1,987.4	19.84 %
	Saving Algorithm	55,800		2,479.6	
4	วิธีหาค่าที่ดีที่สุด	39,200	0 %	1,892	0 %
	Saving Algorithm	39,200		1,892	
3	วิธีหาค่าที่ดีที่สุด	33,000	0.6 %	1,458	0.4 %
	Saving Algorithm	33,200		1,452.2	

จากตารางที่ 3-20 พบว่า กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง มีความแตกต่างของผลลัพธ์ค่าใช้จ่ายแบบเหมาจ่ายทั้งหมดและระยะทางทั้งหมดของการไปรับซากไก่ใน 5 4 และ 3 จุด ประมาณร้อยละ 24 ถึง 19 ไม่ความแตกต่างกัน และร้อยละ 1 ตามลำดับ

จากนั้นจึงทำการแก้ปัญหาเพื่อหาคำตอบตั้งต้นโดยวิธี Saving Algorithm ที่ได้ทำการออกแบบด้วยการใช้ JAVA Script (รหัสคำสั่งโปรแกรมแสดงใน ภาคผนวก ค) โดยผลลัพธ์จากการประมวลผลสำหรับการเดินทางไปรับสินค้าจากจุดของศูนย์รวมต่างๆทั้งหมด 22 จุด จนเต็มกำลังการผลิตของสายการผลิตที่ 1 (30,000 ตันต่อปี) และสายการผลิตที่ 2 (30,000 ตันต่อปี) ดังแสดงในตารางที่ 3-21 ถึง 3-22

ตารางที่ 3-21 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีโรงกำจัดซากตั้งที่ ต. ลำพญากลาง)

รอบการวิ่ง	**เส้นทาง**	**น้ำหนัก (ตัน)**	**ระยะทาง (กม.ต่อวัน)**	**เวลา (นาที)**	**ค่าใช้จ่าย (บาทต่อวัน)**
1	Depot --> สหฟาร์ม (est.129) [7.25] --> Depot	7.25	246	391	6,700
2	Depot --> ซีพีเอฟ (est.03) [5.22] --> โกลเด้น ฟู้ดส์ (est.34) [2.61] --> Depot	7.83	439.9	596.5	9,800
3	Depot --> คาร์กิล มีท (est.32) [6.38] --> เซนทาโก (est.07) [2.61] --> Depot	8.99	383.2	563	8,300
4	Depot --> สหฟาร์ม (est.44) [8.99] --> Depot	8.99	96.4	276.2	3,740
5	Depot --> ซีพีเอฟ (est.23) [6.96] --> ซันฟู้ดส์ (est.25) [2.61] --> Depot	9.57	216.5	407.9	6,700
6	Depot --> ซีพีเอฟ (est.139) [7.54] --> คาร์กิล มีท (est.165) [3.19] --> Depot	10.73	262.2	476.8	6,700
7	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> F&F (est.160) [1.45] --> Depot	11.31	547.6	773.8	11,300
8	Depot --> เบทาโกร (est.49) [7.83] --> แหลมทอง (est.70) [3.77] --> Depot	11.6	313.5	545.5	8,300
9	Depot --> เบทาโกร (est.10) [3.77] --> ไทยโพลทรีกรุ๊ป (est.109) [3.19] --> GFN (est.201) [2.90] --> แหลมทอง (est.06) [2.03] --> Depot	11.89	851.7	1,089.5	16,000
10	Depot --> GFPT (est.11) [3.77] --> พนัส (est.79) [3.77] --> โกลเด้น ฟู้ดส์ (est.53) [2.61] --> ซีพีเอฟ (est.18) [1.74] --> Depot	11.89	792.8	1,030.6	14,400
	รวม	100	4,149.8	6,150.8	91,940

จากตารางที่ 3-21 พบว่า กรณีที่ตั้งโรงกำจัดซากไก่ที่ตำบลลำพญากลาง จังหวัดลพบุรี ไปรับซากและเศษซากไก่จาก 22 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ โดยใช้รถทั้งหมด 20 รอบต่อวัน (โดยแบ่งเป็นการขนส่งซากไก่และเศษซากไก่ทั้งหมด 10 และ 10 รอบต่อวัน ตามลำดับ) ค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายต่อวันประมาณ 183,880 บาท หรือคิดเป็นค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายต่อปีประมาณ 55,164,000 บาท (หากคิดวันทำงานปีละ 300 วัน ดังที่กล่าวในสมมุติฐานข้างต้น) และใช้ระยะทางรวมทั้งหมด 8,299.6 กิโลเมตรต่อวัน ใช้เวลาประมวลผลทั้งหมด 1.2 วินาที เหมือนกันทั้ง 2 กรณี

ตารางที่ 3-22 ผลลัพธ์จาก Saving Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง)

รอบการวิ่ง	**เส้นทาง**	**น้ำหนัก (ตัน)**	**ระยะทาง (กม.ต่อวัน)**	**เวลา (นาที)**	**ค่าใช้จ่าย (บาทต่อวัน)**
1	Depot --> คาร์กิล มีท (est.32) [6.38] --> Depot	6.38	86	213.6	3,740
2	Depot --> ซีพีเอฟ (est.23) [6.96] --> Depot	6.96	29	168.2	3,740
3	Depot --> GFPT (est.11) [3.77] --> ไทยโพลทรี กรุ๊ป (est.109) [3.19] --> Depot	6.96	455	594.2	9,800
4	Depot --> สหฟาร์ม (est.129) [7.25] --> Depot	7.25	382	527	8,300
5	Depot --> เบทาโกร (est.49) [7.83] --> Depot	7.83	109.6	266.2	5,270
6	Depot --> ซีพีเอฟ (est.03) [5.22] --> เบทาโกร (est.10) [3.77] --> Depot	8.99	350.6	530.4	8,300
7	Depot --> พนัส (est.79) [3.77] --> GFN (est.201) [2.90] --> แหลมทอง (est.06) [2.03] --> ซีพีเอฟ (est.18) [1.74] --> Depot	10.44	616.9	825.7	13,000
8	Depot --> ซีพีเอฟ (est.139) [7.54] --> คาร์กิล มีท (est.165) [3.19] --> Depot	10.73	326.2	540.8	8,300
9	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> F&F (est.160) [1.45] --> Depot	11.31	387.6	613.8	8,300
10	Depot --> สหฟาร์ม (est.44) [8.99] --> โกลเด้น ฟู้ด (est.53) [2.61] --> Depot	11.6	319.9	551.9	8,300
11	Depot --> แหลมทอง (est.70) [3.77] --> โกลเด้น ฟู้ดส์ (est.34) [2.61] --> ซันฟู้ดส์ (est.25) [2.61] --> เซนทาโก (est.07) [2.61] --> Depot	11.6	706	938	14,400
	รวม	100	3,768.8	5,769.8	91,450

จากตารางที่ 3-22 พบว่า กรณีที่ตั้งโรงกำจัดซากไก่ที่ตำบลทับกวาง จังหวัดสระบุรี ไปรับซากและเศษซากไก่จาก 22 จุด โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ โดยใช้รถทั้งหมด 22 รอบต่อวัน (โดยแบ่งเป็นการขนส่งซากไก่และเศษซากไก่ทั้งหมด 11 และ 11 รอบต่อวัน ตามลำดับ) ค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายต่อวันประมาณ 182,900 บาท หรือคิดเป็นค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายต่อปีประมาณ 54,870,000 บาท (หากคิดวันทำงานปีละ 300 วัน ดังที่กล่าวในสมมุติฐานข้างต้น) และใช้ระยะทางรวมทั้งหมด 7,537.6 กิโลเมตรต่อวัน ใช้เวลาประมวลผลทั้งหมด 1.1 วินาที เหมือนกันทั้ง 2 กรณี

ต่อไปจะเป็นการปรับปรุงคำตอบตั้งต้นที่ได้จาก Saving Algorithm โดย Genetic Algorithm ซึ่งจะกล่าวต่อไปในบทที่ 4

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 4
การปรับปรุงเส้นทางโดยวิธีการเชิงพันธุกรรม

หลังจากที่ได้เส้นทางการขนส่งซึ่งเป็นคำตอบตั้งต้นโดย Saving Algorithm แล้ว ในระยะที่สองผู้วิจัยจะทำการปรับปรุงคำตอบตั้งต้นที่ได้โดย Genetic Algorithm โดยในบทนี้จะประกอบไปด้วยขั้นตอนของการปรับปรุงคำตอบ การตั้งค่าและการทดลองค่าพารามิเตอร์ต่างๆ เพื่อให้ได้ผลลัพธ์การจัดเส้นทางที่มีต้นทุนค่าใช้จ่ายการขนส่งแบบเหมาจ่ายและระยะทางรวมที่น้อยที่สุด

4.1 ขั้นตอนของ Genetic Algorithm (ตัวอย่างการปรับปรุงแสดงใน ภาคผนวก ก) มีดังนี้

1. การสร้างโครโมโซมแทนคำตอบจากคำตอบตั้งต้น
2. การสร้างโครโมโซมแทนคำตอบเพิ่ม
3. พิจารณาค่าความเหมาะสม (Fitness) ของประชากรตั้งต้น
4. ทำการคัดเลือกโครโมโซมโดยการใช้การสร้าง Roulette Wheel
5. ทำการคัดเลือกโครโมโซมด้วยวิธีการ Binary Tournament Selection
6. ทำการคัดเลือกโครโมโซมเข้าสู่ Mating Pool เพื่อเข้าสู่ Genetic Algorithm
7. ทำการทดลองเพื่อหาจำนวนรอบการทดสอบ (replication) ที่เหมาะสม
8. ทำการทดลองความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm) ที่เหมาะสมในงานวิจัย
9. เข้าสู่ Genetic Algorithm โดยใช้พารามิเตอร์ตามค่าที่ได้ทำการทดลอง

4.1.1 การสร้างโครโมโซมแทนคำตอบจากคำตอบตั้งต้น

การสร้างโครโมโซมแทนคำตอบจากคำตอบตั้งต้น เพื่อใช้เป็นโครโมโซมพ่อและแม่ตั้งต้นใน Genetic Algorithm โดยมีการจัดเซตของแต่ละยีนส์เป็น (ลำดับรอบการวิ่ง, ลำดับของลูกค้าภายในลำดับรอบการวิ่งนั้น) ตัวอย่างเช่น {(1,2),(2,2),(1,1),(2,1)} หมายถึง จุดของลูกค้าที่ 1 อยู่ในลำดับรอบการวิ่งที่ 1 ลำดับภายในที่ 2 และ จุดของลูกค้าที่ 2 อยู่ในลำดับรอบการวิ่งที่ 2 ลำดับภายในที่ 2 เป็นต้น

4.1.2 การสร้างโครโมโซมแทนคำตอบเพิ่ม

การสร้างโครโมโซมแทนคำตอบเพิ่ม โดยทำการสลับยีนส์ภายในแต่ละลำดับรอบการวิ่งทั้งหมด และสลับยีนส์ระหว่างแต่ละรอบการวิ่ง โดยเงื่อนไขการสลับยีนส์ระหว่างแต่ละรอบการวิ่งมีวิธีการ คือ เลือกลำดับรอบการวิ่งที่ภายในมีจำนวนจุดของลูกค้า 2 และ 3 จุด แล้วนำแต่ละจุดของลูกค้าในลำดับรอบการวิ่งที่ได้เลือกนั้นไปทำการสลับกับจุดของลูกค้าอื่นๆที่ภายในลำดับรอบการวิ่งมีจำนวนจุดของลูกค้า 2 และ 3 จุด เหมือนกัน ทำการสลับตามเงื่อนไขจนครบทุกกรณี จากนั้นจะได้โครโมโซมตั้งต้นใหม่เพิ่มอีก 99 โครโมโซม แล้วทำการรวมโครโมโซมตั้งต้นเดิมและใหม่รวมเป็น 100 โครโมโซม เพื่อนำมาเป็นประชากรตั้งต้น

Chulalongkorn University

4.1.3 พิจารณาค่าความเหมาะสม (Fitness) ของประชากรตั้งต้น

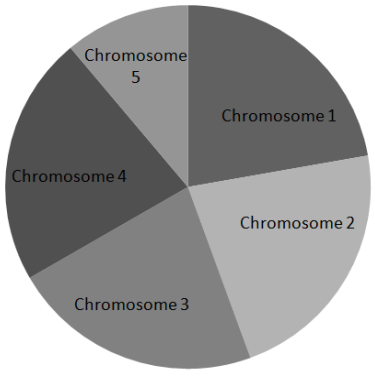
พิจารณาค่าความเหมาะสม (Fitness) ของประชากรตั้งต้น โดยพิจารณาจากสมการวัตถุประสงค์โดยใช้ค่าใช้จ่ายแบบเหมารวมของทุกรอบการวิ่ง โดยกำหนดให้ค่าความเหมาะสมของโครโมโซมที่มีค่าใช้จ่ายน้อยที่สุด มีค่ามากที่สุด ตามลำดับ

4.1.4 ทำการคัดเลือกโครโมโซมโดยการใช้การสร้าง Roulette Wheel

ทำการคัดเลือกโครโมโซมโดยการใช้การสร้าง Roulette Wheel โดยหาค่าผลรวมของ Fitness ของโครโมโซมทั้งหมด แล้วจึงหาค่าความน่าจะเป็นในการถูกคัดเลือก (Probability of Selection : p_i) และความน่าจะเป็นในการถูกคัดเลือกสะสม (Cumulative Probability of Selection : q_i) ของโครโมโซมแต่ละตัว โดยตัวอย่างการคัดเลือกโครโมโซมโดยการใช้การสร้าง Roulette Wheel ดังแสดงในตารางที่ 4-1 และรูปที่ 4-1

ตารางที่ 4-1 ตัวอย่างการสร้าง Roulette Wheel

โครโมโซม	**ค่าความเหมาะสม**	p_i	q_i
1	2	0.222	0.222
2	2	0.222	0.444
3	2	0.222	0.666
4	2	0.222	0.888
5	1	0.111	1
	9		



รูปที่ 4-1 ตัวอย่างการสร้าง Roulette Wheel

4.1.5 ทำการคัดเลือกโครโมโซมด้วยวิธีการ Binary Tournament Selection

ทำการคัดเลือกโครโมโซมด้วยวิธีการ Binary Tournament Selection โดยการสร้างตัวเลขสุ่ม (r) ที่มีค่าระหว่าง 0 ถึง 1 ออกมาสองค่า คือ เมื่อสุ่มโครโมโซมสองตัวออกมาแล้วให้นำมาพิจารณาค่าความเหมาะสมของโครโมโซมแต่ละตัว ถ้าโครโมโซมใดมีค่าความเหมาะสมมากกว่าให้เลือกโครโมโซมนั้น แต่ถ้าทั้งสองโครโมโซมมีค่าความเหมาะสมเท่ากัน ให้พิจารณาจากค่า ค่าใช้จ่ายการขนส่งแบบเหมาจ่ายรวม ถ้าโครโมโซมใดมีค่า ค่าใช้จ่ายการขนส่งแบบเหมาจ่ายน้อยกว่าให้เลือก

โครโมโซมนั้น แต่ถ้าทั้งสองโครโมโซมมีค่า ค่าใช้จ่ายการขนส่งแบบเหมาจ่ายเท่ากัน ให้พิจารณาจากค่าระยะทางรวม ถ้าโครโมโซมใดมีค่า ระยะทางรวม น้อยกว่าให้เลือกโครโมโซมนั้น แต่ถ้าทั้งสองโครโมโซมมีค่าระยะทางรวมเท่ากันอีก ให้สุ่มเลือกโครโมโซม 1 โครโมโซม แล้วจึงนำเข้าสู่ Mating Pool โดยตัวอย่างการคัดเลือกโครโมโซมด้วยวิธีการ Binary Tournament Selection ดังแสดงในตารางที่ 4-2

ตารางที่ 4-2 ตัวอย่างการคัดเลือกโครโมโซมด้วยวิธีการ Binary Tournament Selection

ลำดับ	ประชากรที่ 1				ประชากรที่ 2				ลำดับโครโมโซมที่คัดเลือก
	r_1	$r_1 < q_i$	โครโมโซม	ค่าความเหมาะสม	r_2	$r_2 < q_i$	โครโมโซม	ค่าความเหมาะสม	
1	0.7894	0.888	4	2	0.9843	1	5	1	4
2	0.5678	0.666	3	2	0.8457	0.888	4	2	3
3	0.4789	0.666	3	2	0.3246	0.444	2	2	1
4	0.2938	0.444	2	2	0.6742	0.888	4	2	2
5	0.1123	0.222	1	2	0.5213	0.666	3	2	1

4.1.6 ทำการคัดเลือกโครโมโซมเข้าสู่ Mating Pool เพื่อเข้าสู่ Genetic Algorithm

เมื่อทำการคัดเลือกโครโมโซมเข้าสู่ Mating Pool แล้วจึงเข้าสู่ Genetic Algorithm โดยได้กำหนดค่าพารามิเตอร์ไว้ ดังแสดงในตารางที่ 4-3

ตารางที่ 4-3 ค่าพารามิเตอร์ของ Genetic Algorithm สำหรับการจัดเส้นทางของโรงกำจัดซากไก่

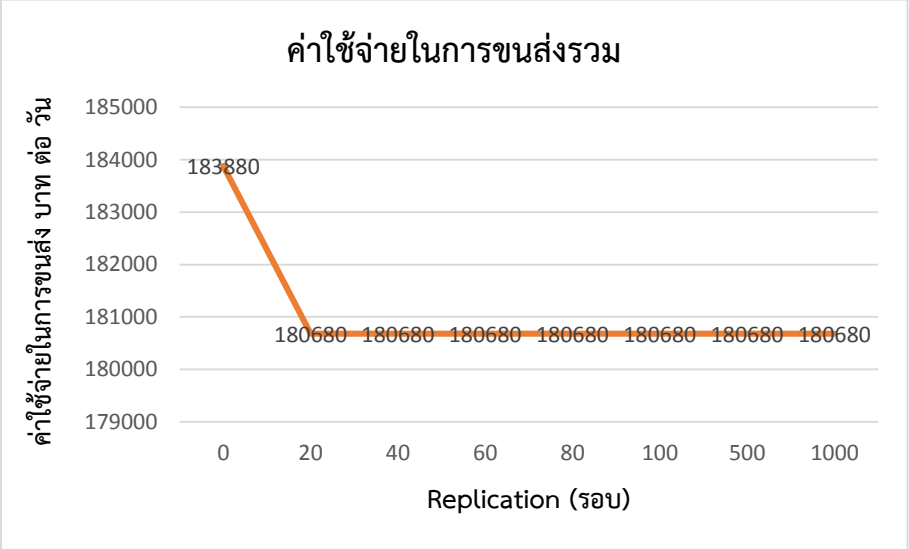
พารามิเตอร์	ค่าพารามิเตอร์	อ้างอิงจาก
วิธีการครอสโอเวอร์	Weighted Mapping Crossover	[12]
วิธีการมิวเตชั่น	Reciprocal Exchange Mutation	[12]
ความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm)	Set 1 : Pc=0.7 , Pm=0.1	การทดลอง
	Set 2 : Pc=0.7 , Pm=0.2	
	Set 3 : Pc=0.7 , Pm=0.3	
	Set 4 : Pc=0.8 , Pm=0.1	
	Set 5 : Pc=0.8 , Pm=0.2	
	Set 6 : Pc=0.8 , Pm=0.3	
	Set 7 : Pc=0.9 , Pm=0.1	
	Set 8 : Pc=0.9 , Pm=0.2	
	Set 9 : Pc=0.9 , Pm=0.3	
จำนวนรอบการทดสอบ (replication)	20 รอบ	การทดลอง

4.1.7 ทำการทดลองเพื่อหาจำนวนรอบการทดสอบ (replication) ที่เหมาะสม

ทำการทดลองเพื่อหาจำนวนรอบการทดสอบ (replication) ที่เหมาะสม โดยเริ่มต้นทดสอบที่ 20 รอบจนถึง 1000 รอบ โดยขยับเพิ่มครั้งละ 20 รอบ และทำการสุ่มใช้ความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm) ที่ 0.7 และ 0.3 ตามลำดับ โดยแบ่งการทดลองทั้งหมดออกเป็น 2 กรณี โดยผลการทดลอง ดังแสดงในรูปที่ 4-2 ถึง 4-3 ได้แก่

- การเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง
- การเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง

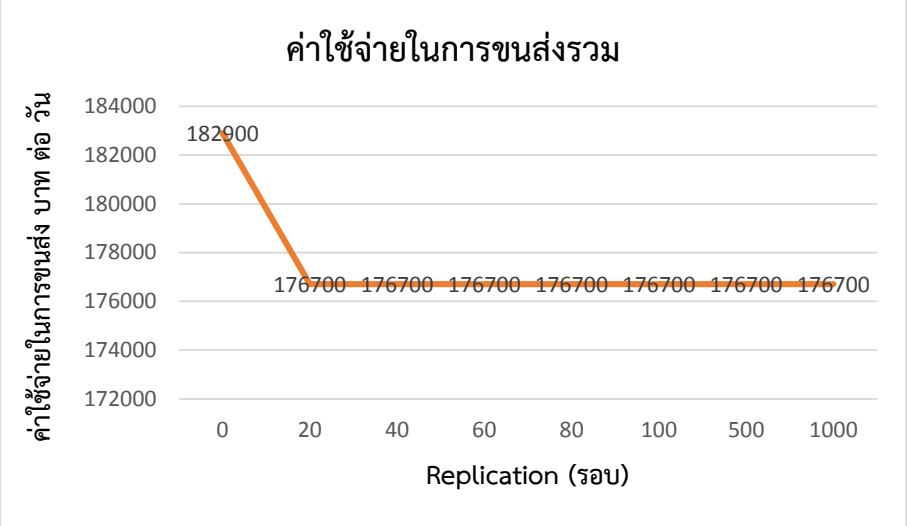
4.1.7.1 การเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง



รูปที่ 4-2 แสดงผลของราคาค่าขนส่งแบบเหมาจ่ายรวมต่อจำนวนรอบการทดลอง (สำหรับการเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง)

จากรูปที่ 4-2 ผลการทดลองการหาจำนวนรอบของการทดสอบที่เหมาะสม พบว่าค่าใช้จ่ายแบบเหมาจ่ายรวมในการทดสอบที่ 20 รอบ จนถึง 1000 รอบ ผลที่ได้ไม่แตกต่างกัน คือ 180,680 บาท ดังนั้นจึงกำหนดจำนวนรอบการทดสอบที่ 20 รอบ ในทุกๆการทดลองความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm) สำหรับการเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง

4.1.7.2 การเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง



รูปที่ 4-3 แสดงผลของราคาค่าขนส่งแบบเหมาจ่ายรวมต่อจำนวนรอบการทดลอง
(สำหรับการเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง)

จากรูปที่ 4-3 ผลการทดลองการหาจำนวนรอบของการทดสอบที่เหมาะสม พบว่าค่าใช้จ่ายแบบเหมาจ่ายรวมในการทดสอบที่ 20 รอบ จนถึง 1000 รอบ ผลที่ได้ไม่แตกต่างกัน คือ 176,700 บาท ดังนั้นจึงกำหนดจำนวนรอบการทดสอบที่ 20 รอบ ในทุกๆการทดลองความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm) สำหรับการเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง

4.1.8 ทำการทดลองความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm) ที่เหมาะสมในงานวิจัย

เมื่อได้จำนวนรอบของการทดสอบที่เหมาะสมในงานวิจัยแล้ว จึงทำการทดลองความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm) ที่เหมาะสมในงานวิจัย โดยแบ่งการทดลองออกเป็น 9 เซต โดยในแต่ละเซตมีการเก็บข้อมูลของค่าใช้จ่ายแบบเหมาจ่ายรวม ระยะทางรวม และเวลาในการประมวลผลของแต่ละเซต เซตละ 2 ครั้ง และแบ่งการทดลองทั้งหมดออกเป็น 2 กรณี ได้แก่

- การเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง
- การเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง

4.1.8.1 การเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง

คำตอบที่ได้จากผลการทดลองที่เก็บได้จากการประมวลผลทั้ง 18 ครั้ง ได้ค่าเหมือนกัน คือ มีค่าใช้จ่ายแบบเหมาจ่ายรวม 180,680 บาท และ มีระยะทางรวม 8,129.6 กม. แตกต่างเพียงเวลาในการประมวลผลเพียงเล็กน้อย ดังแสดงในตารางที่ 4-4

ตารางที่ 4-4 ผลการทดลองความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm)
(สำหรับการเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง)

ลำดับเซต	ลำดับครั้ง	Pc	Pm	ค่าใช้จ่าย (บาทต่อวัน)	ระยะทาง (กม.ต่อวัน)	เวลาในการการประมวลผล (วินาที)
1	1	0.7	0.1	180,680	8,129.6	10.87
	2			180,680	8,129.6	10.91
2	1	0.7	0.2	180,680	8,129.6	10.89
	2			180,680	8,129.6	10.78
3	1	0.7	0.3	180,680	8,129.6	10.79
	2			180,680	8,129.6	10.81
4	1	0.8	0.1	180,680	8,129.6	10.80
	2			180,680	8,129.6	10.75
5	1	0.8	0.2	180,680	8,129.6	10.81
	2			180,680	8,129.6	10.85
6	1	0.8	0.3	180,680	8,129.6	10.86
	2			180,680	8,129.6	10.80
7	1	0.9	0.1	180,680	8,129.6	10.77
	2			180,680	8,129.6	10.85
8	1	0.9	0.2	180,680	8,129.6	10.71
	2			180,680	8,129.6	10.87
9	1	0.9	0.3	180,680	8,129.6	10.84
	2			180,680	8,129.6	10.77

จากนั้นจึงได้นำผลการทดลองข้างต้นไปประมวลผลโดยใช้โปรแกรม minitab16 เพื่อวิเคราะห์ค่าความแปรปรวนของข้อมูลที่ได้ทำการเก็บเพื่อหาค่าพารามิเตอร์ของความน่าจะเป็นของการครอสโอเวอร์ (Pc) และ มิวเตชั่น (Pm) ที่เหมาะสม ดังแสดงในรูปที่ 4-4

One-way ANOVA: Cost versus Variable

```
Source    DF         SS         MS  F  P
Variable   8  0.0000000  0.0000000  *  *
Error      9  0.0000000  0.0000000
Total     17  0.0000000

S = 0   R-Sq = *%   R-Sq(adj) = *%


                          Individual 95% CIs For Mean Based on Pooled StDev
Level  N    Mean  StDev       +---------+---------+---------+---------
1      2  180680      0       *
2      2  180680      0       *
3      2  180680      0       *
4      2  180680      0       *
5      2  180680      0       *
6      2  180680      0       *
7      2  180680      0       *
8      2  180680      0       *
9      2  180680      0       *
                               +---------+---------+---------+---------
                          180680    180700    180720    180740

Pooled StDev = 0
```

One-way ANOVA: Distance versus Variable

```
Source    DF         SS         MS  F  P
Variable   8  0.0000000  0.0000000  *  *
Error      9  0.0000000  0.0000000
Total     17  0.0000000

S = 0   R-Sq = *%   R-Sq(adj) = *%


                          Individual 95% CIs For Mean Based on
                          Pooled StDev
Level  N     Mean  StDev   ----+---------+---------+---------+-----
1      2  8129.60   0.00   *
2      2  8129.60   0.00   *
3      2  8129.60   0.00   *
4      2  8129.60   0.00   *
5      2  8129.60   0.00   *
6      2  8129.60   0.00   *
7      2  8129.60   0.00   *
8      2  8129.60   0.00   *
9      2  8129.60   0.00   *
                          ----+---------+---------+---------+-----
                          8130.0    8131.0    8132.0    8133.0

Pooled StDev = 0.00
```

รูปที่ 4-4 แสดงการวิเคราะห์ค่าความแปรปรวนแบบทางเดียว (One way Anova) ระหว่างค่าใช้จ่ายรวมกับข้อมูลแต่ละครั้งและระหว่างระยะทางรวมกับข้อมูลแต่ละครั้ง (สำหรับการเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง)

จากรูปที่ 4-4 การวิเคราะห์ค่าความแปรปรวนแบบทางเดียว โดยสังเกตจากค่าของ P-Value พบว่า ค่าใช้จ่ายแบบเหมาจ่ายรวมและระยะทางรวมของในแต่ละเซตไม่มีความแตกต่างกันอย่างมีนัยสำคัญที่ระดับความเชื่อมั่นที่ 95% ดังนั้นจึงต้องใช้เวลาในการประมวลผลของแต่ละเซตในการคัดเลือก โดยเซตที่ 4 Pc=0.8, Pm=0.1 ใช้เวลาในการประมวลผลเร็วที่สุด คือ 10.75 วินาที ดังนั้นจึงเลือกใช้พารามิเตอร์ความน่าจะเป็นในการครอสโอเวอร์ที่ 0.8 และมิวเตชั่นที่ 0.1

4.1.8.2 การเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง

คำตอบที่ได้จากผลการทดลองที่เก็บได้จากการประมวลผลทั้ง 18 ครั้ง ได้ค่าเหมือนกัน คือ มีค่าใช้จ่ายแบบเหมาจ่ายรวม 176,700 บาท และ มีระยะทางรวม 7,201 กม. แตกต่างเพียงเวลาในการประมวลผลเพียงเล็กน้อย ดังแสดงในตารางที่ 4-5

ตารางที่ 4-5 ผลการทดลองความน่าจะเป็นในการครอสโอเวอร์ (Pc) และมิวเตชั่น (Pm)
(สำหรับการเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง)

ลำดับเซต	ลำดับครั้ง	Pc	Pm	ค่าใช้จ่าย (บาทต่อวัน)	ระยะทาง (กม.ต่อวัน)	เวลาในการการประมวลผล (วินาที)
1	1	0.7	0.1	176,700	7,201	10.90
	2			176,700	7,201	10.92
2	1	0.7	0.2	176,700	7,201	10.87
	2			176,700	7,201	10.89
3	1	0.7	0.3	176,700	7,201	10.88
	2			176,700	7,201	10.86
4	1	0.8	0.1	176,700	7,201	10.85
	2			176,700	7,201	10.89
5	1	0.8	0.2	176,700	7,201	10.82
	2			176,700	7,201	10.81
6	1	0.8	0.3	176,700	7,201	10.88
	2			176,700	7,201	10.87
7	1	0.9	0.1	176,700	7,201	10.94
	2			176,700	7,201	10.92
8	1	0.9	0.2	176,700	7,201	10.95
	2			176,700	7,201	10.93
9	1	0.9	0.3	176,700	7,201	10.96
	2			176,700	7,201	10.97

จากนั้นจึงได้นำผลการทดลองข้างต้นไปประมวลผลโดยใช้โปรแกรม minitab16 เพื่อวิเคราะห์ค่าความแปรปรวนของข้อมูลที่ได้ทำการเก็บเพื่อหาค่าพารามิเตอร์ของความน่าจะเป็นของการครอสโอเวอร์ (Pc) และ มิวเตชั่น (Pm) ที่เหมาะสม ดังแสดงในรูปที่ 4-5

One-way ANOVA: Cost versus Variable

```
Source    DF         SS         MS  F  P
Variable   8  0.0000000  0.0000000  *  *
Error      9  0.0000000  0.0000000
Total     17  0.0000000

S = 0   R-Sq = *%   R-Sq(adj) = *%


                          Individual 95% CIs For Mean Based on Pooled StDev
Level  N    Mean  StDev       +---------+---------+---------+---------
1      2  176700      0       *
2      2  176700      0       *
3      2  176700      0       *
4      2  176700      0       *
5      2  176700      0       *
6      2  176700      0       *
7      2  176700      0       *
8      2  176700      0       *
9      2  176700      0       *
                          +---------+---------+---------+---------
                       176700    176720    176740    176760

Pooled StDev = 0
```

One-way ANOVA: Distance versus Variable

```
Source    DF         SS         MS  F  P
Variable   8  0.0000000  0.0000000  *  *
Error      9  0.0000000  0.0000000
Total     17  0.0000000

S = 0   R-Sq = *%   R-Sq(adj) = *%


                          Individual 95% CIs For Mean Based on
                          Pooled StDev
Level  N     Mean  StDev  --------+---------+---------+---------+-
1      2  7201.00   0.00  *
2      2  7201.00   0.00  *
3      2  7201.00   0.00  *
4      2  7201.00   0.00  *
5      2  7201.00   0.00  *
6      2  7201.00   0.00  *
7      2  7201.00   0.00  *
8      2  7201.00   0.00  *
9      2  7201.00   0.00  *
                          --------+---------+---------+---------+-
                             7201.60   7202.40   7203.20   7204.00

Pooled StDev = 0.00
```

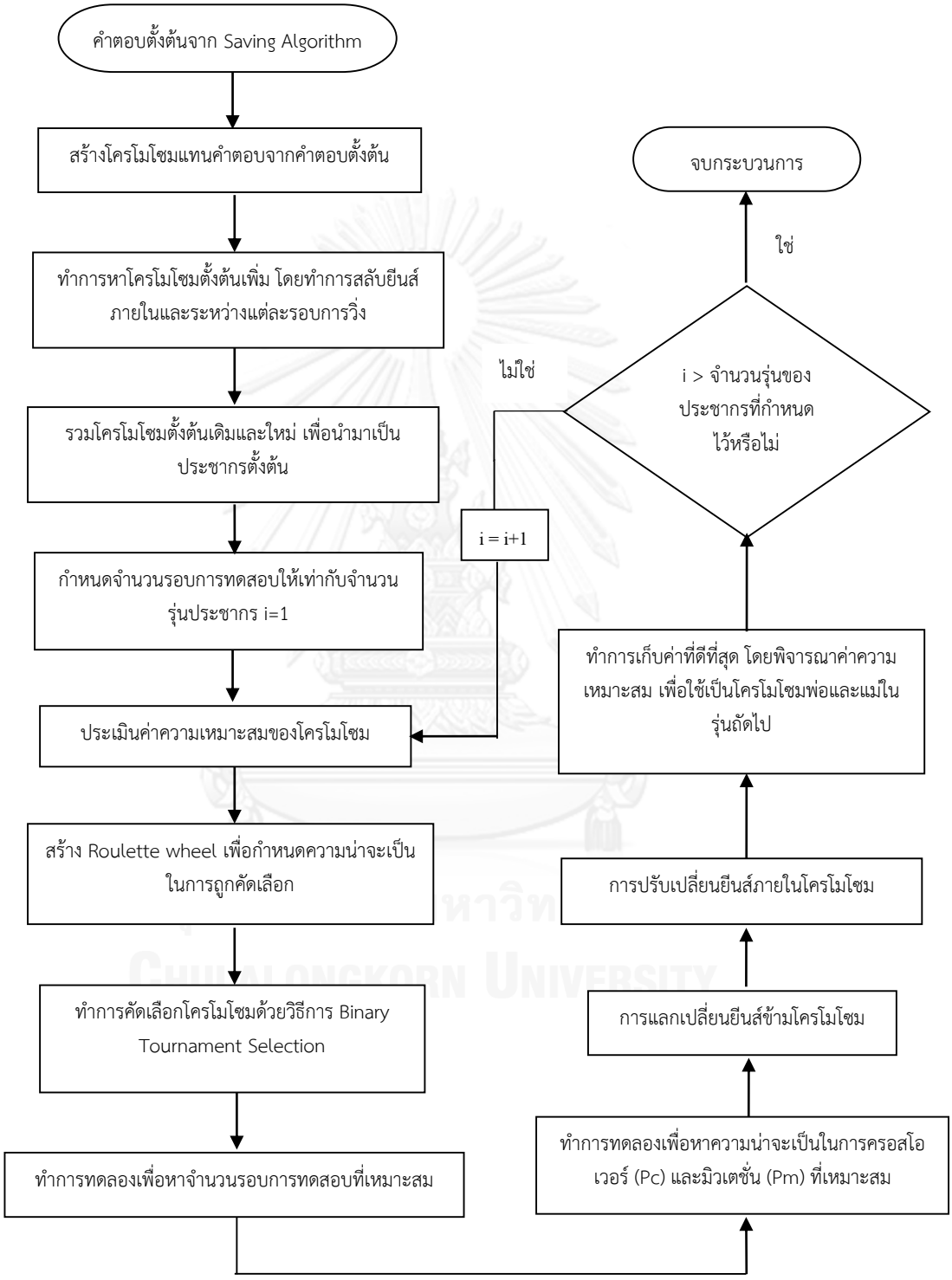
รูปที่ 4-5 แสดงการวิเคราะห์ค่าความแปรปรวนแบบทางเดียว (One way Anova) ระหว่างค่าใช้จ่าย รวมกับข้อมูลแต่ละครั้งและระหว่างระยะทางรวมกับข้อมูลแต่ละครั้ง (สำหรับการเดินทางไปรับซากและเศษซากไก่กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง)

จากรูปที่ 4-5 การวิเคราะห์ค่าความแปรปรวนแบบทางเดียว โดยสังเกตจากค่าของ P-Value พบว่า ค่าใช้จ่ายแบบเหมาจ่ายรวมและระยะทางรวมของในแต่ละเซตไม่มีความแตกต่างกันอย่างมีนัยสำคัญที่ระดับความเชื่อมั่นที่ 95% ดังนั้นจึงต้องใช้เวลาในการประมวลผลของแต่ละเซตในการคัดเลือก โดยเซตที่ 5 Pc=0.8, Pm=0.2 ใช้เวลาในการประมวลผลเร็วที่สุด คือ 10.82 วินาที ดังนั้นจึงเลือกใช้พารามิเตอร์ความน่าจะเป็นในการครอสโอเวอร์ที่ 0.8 และมิวเตชั่นที่ 0.2

4.1.9 เข้าสู่ Genetic Algorithm โดยใช้พารามิเตอร์ตามค่าที่ได้ทำการทดลอง

เมื่อทำได้ทำการทดลองค่าพารามิเตอร์ต่างๆแล้วจึงทำการปรับปรุงคำตอบตั้งต้นโดย Genetic Algorithm ตามค่าพารามิเตอร์ที่ได้ทำการทดลองไว้ข้างต้น

โดยขั้นตอนการปรับปรุงคำตอบตั้งต้นโดย Genetic Algorithm สามารถแสดงเป็นแผนผัง ดังแสดงในรูปที่ 4-6



รูปที่ 4-6 แผนผังขั้นตอนการปรับปรุงคำตอบตั้งต้นโดย Genetic Algorithm

4.2 ผลการวิจัย

ในงานวิจัยนี้จะทำการปรับปรุงคำตอบตั้งต้นโดย Genetic Algorithm โดยทำการออกแบบด้วยการใช้ JAVA Script (รหัสคำสั่งโปรแกรมแสดงใน ภาคผนวก ค) และทดสอบบนเครื่องคอมพิวเตอร์ รุ่น Core (TM) 2 Duo Processor P8600 2.4 Ghz, 3.0 GB of RAM บน Windows XP โดยได้ผลลัพธ์จากการประมวลผล ดังแสดงในตารางที่ 4-6 ถึง 4-7

ตารางที่ 4-6 ผลลัพธ์จาก Genetic Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีโรงกำจัดซากตั้งที่ ต.ลำพญากลาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot --> สหฟาร์ม (est.129) [7.25] --> Depot	7.25	246	391	6,700
2	Depot --> ซีพีเอฟ (est.03) [5.22] --> โกลเด้น ฟู้ดส์ (est.34) [2.61] --> Depot	7.83	439.9	596.5	9,800
3	Depot --> คาร์กิล มีท (est.32) [6.38] --> เซนทาโก (est.07) [2.61] --> Depot	8.99	383.2	563	8,300
4	Depot --> สหฟาร์ม (est.44) [8.99] --> Depot	8.99	96.4	276.2	3,740
5	Depot --> ซีพีเอฟ (est.23) [6.96] --> ซันฟู้ดส์ (est.25) [2.61] --> Depot	9.57	216.5	407.9	6,700
6	Depot --> ซีพีเอฟ (est.139) [7.54] --> คาร์กิล มีท (est.165) [3.19] --> Depot	10.73	262.2	476.8	6,700
7	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> F&F (est.160) [1.45] --> Depot	11.31	547.6	773.8	11,300
8	Depot --> เบทาโกร (est.49) [7.83] --> แหลมทอง (est.70) [3.77] --> Depot	11.6	313.5	545.5	8,300
9	Depot --> GFN (est.201) [2.90] --> ไทยโพลทรีกรุ๊ป (est.109) [3.19] --> เบทาโกร (est.10) [3.77] --> แหลมทอง (est.06) [2.03] --> Depot	11.89	766.7	1,004.5	14,400
10	Depot --> GFPT (est.11) [3.77] --> พนัส (est.79) [3.77] --> โกลเด้น ฟู้ดส์ (est.53) [2.61] --> ซีพีเอฟ (est.18) [1.74] --> Depot	11.89	792.8	1,030.6	14,400
	รวม	100	4,064.8	6,065.8	90,340

ตารางที่ 4-7 ผลลัพธ์จาก Genetic Algorithm โดยรับซากและเศษซากไก่ทั้งหมด 22 จุด (กรณีโรงกำจัดซากตั้งที่ ต.ทับกวาง)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot --> คาร์กิล มีท (est.32) [6.38] --> Depot	6.38	86	213.6	3,740
2	Depot --> ซีพีเอฟ (est.23) [6.96] --> Depot	6.96	29	168.2	3,740
3	Depot --> GFPT (est.11) [3.77] --> ไทยโพลทรีกรุ๊ป (est.109) [3.19] --> Depot	6.96	455	594.2	9,800
4	Depot --> สหฟาร์ม (est.129) [7.25] --> Depot	7.25	382	527	8,300
5	Depot --> เบทาโกร (est.49) [7.83] --> Depot	7.83	109.6	266.2	5,270
6	Depot --> ซีพีเอฟ (est.03) [5.22] --> เบทาโกร (est.10) [3.77] --> Depot	8.99	350.6	530.4	8,300
7	Depot --> พนัส (est.79) [3.77] --> GFN (est.201) [2.90] --> แหลมทอง (est.06) [2.03] --> ซีพีเอฟ (est.18) [1.74] --> Depot	10.44	616.9	825.7	13,000
8	Depot --> ซีพีเอฟ (est.139) [7.54] --> คาร์กิล มีท (est.165) [3.19] --> Depot	10.73	326.2	540.8	8,300
9	Depot --> ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> F&F (est.160) [1.45] --> Depot	11.31	387.6	613.8	8,300
10	Depot --> สหฟาร์ม (est.44) [8.99] --> โกลเด้น ฟู้ดส์ (est.53) [2.61] --> Depot	11.6	319.9	551.9	8,300
11	Depot --> แหลมทอง (est.70) [3.77] --> ซันฟู้ดส์ (est.25) [2.61] --> โกลเด้น ฟู้ดส์ (est.34) [2.61] --> เซนทาโก (est.07) [2.61] --> Depot	11.6	537.7	769.7	11,300
	รวม	100	3,600.5	5,601.5	88,350

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การจัดเส้นทางไปรับซากและเศษซากไก่โดยจะทำการแยกรถไม่ให้ใช้ร่วมกันเพื่อสอดรับกับเกณฑ์ในการเคลื่อนย้ายซากไก่ กรณีที่ตั้งโรงกำจัดซากไก่ที่ตำบลลำพญากลาง จังหวัดลพบุรี พบว่า ใช้รถทั้งหมด 20 รอบต่อวัน (โดยแบ่งเป็นการขนส่งซากไก่และเศษซากไก่ทั้งหมด 10 และ 10 รอบต่อวัน ตามลำดับ) ค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมต่อวันประมาณ 180,680 บาท หรือคิดเป็นค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมต่อปีประมาณ 54,204,000 บาท (หากคิดวันทำงานปีละ 300 วัน ดังที่กล่าวในสมมุติฐานข้างต้น) และใช้ระยะทางรวมทั้งหมด 8,129.6 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 10.75 วินาที และ กรณีที่ตั้งโรงกำจัดซากไก่ที่ตำบลทับกวาง จังหวัดสระบุรี พบว่า ใช้รถทั้งหมด 22 รอบต่อวัน (โดยแบ่งเป็นการขนส่งซากไก่และเศษซากไก่ทั้งหมด 11 และ 11 รอบต่อวัน ตามลำดับ) ค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมต่อวันประมาณ 176,700 บาท หรือคิดเป็นค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมต่อปีประมาณ 53,010,000 บาท (หากคิดวันทำงานปีละ 300 วัน ดังที่กล่าวในสมมุติฐานข้างต้น) และใช้ระยะทางรวมทั้งหมด 7,201 กิโลเมตรต่อวัน โดยใช้เวลาประมวลผลทั้งหมด 10.82 วินาที

การพัฒนาโดยการปรับปรุงคำตอบตั้งต้นโดย Genetic Algorithm เมื่อทำการเปรียบเทียบกับ Saving Algorithm สามารถลดค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมต่อปีและระยะทางรวมต่อปี ดังแสดงในตารางที่ 4-8

ตารางที่ 4-8 เปรียบเทียบผลลัพธ์ค่าใช้จ่ายแบบเหมาจ่ายรวมและระยะทางรวม ของกรณีโรงกำจัดซากตั้งที่ ต. ลำพญากลาง และ ต.ทับกวาง

สถานที่ของโรงกำจัดซากไก่	วิธีการ	ค่าใช้จ่าย (บาทต่อปี)	ความแตกต่างของค่าใช้จ่าย	ระยะทาง (กม.ต่อปี)	ความแตกต่างของระยะทาง
ต.ลำพญากลาง	Saving Algorithm	55,164,000	1.74 %	2,489,880	2.05 %
	Genetic Algorithm	54,204,000		2,438,880	
ต.ทับกวาง	Saving Algorithm	54,870,000	3.39 %	2,261,280	4.47 %
	Genetic Algorithm	53,010,000		2,160,300	

จากตารางที่ 4-8 พบว่า กรณีโรงกำจัดซากไก่ตั้งที่ ต.ลำพญากลาง Genetic Algorithm สามารถลดค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมและระยะทางรวมได้ 960,000 บาทต่อปี และ 51,000 กม.ต่อปี ตามลำดับ หรือ ร้อยละ 1.74 และ 2.05 ตามลำดับ เมื่อเปรียบเทียบกับ Saving Algorithm และในกรณีโรงกำจัดซากไก่ตั้งที่ ต.ทับกวาง Genetic Algorithm สามารถลดค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมและระยะทางรวมได้ 1,860,000 บาทต่อปี และ 100,980 กม.ต่อปี ตามลำดับ หรือ ร้อยละ 3.39 และ 4.47 ตามลำดับ เมื่อเปรียบเทียบกับ Saving Algorithm

บทที่ 5
การสรุปผลการวิจัย และ ข้อเสนอแนะ

หลังจากที่ได้ทำการหาคำตอบเส้นทางตั้งต้นและปรับปรุงเส้นทางแล้ว ต่อไปเป็นการสรุปผลการวิจัยทั้งหมดพร้อมกับข้อเสนอแนะของงานวิจัย

5.1 สรุปผลการวิจัย

ในงานวิจัยนี้มีจุดมุ่งหมายที่จะสร้างเส้นทางการขนส่งซากและเศษซากไก่จากลูกค้าทั้งหมด 22 จุด ไปยังโรงกำจัดซากไก่ โดยโรงกำจัดซากไก่มีปริมาณกำลังการผลิตของสายการผลิตทั้ง 2 สาย เท่ากับ 60,000 ตันต่อปี ซึ่งเมื่อคำนวณปริมาณทั้งหมดแล้วพบว่าจะมีปริมาณเกินกำลังการผลิตของสายการผลิตทั้ง 2 สาย จึงปรับปริมาณวัตถุดิบที่จะไปรับทั้ง 22 จุด โดยการปรับตามอัตราส่วน)proportional) เพื่อให้สามารถรองรับกำลังการผลิตได้ โดยมีตัวอย่างของตำแหน่งที่ตั้งอยู่ แห่ง 2 (1 คือบริเวณตำบลลำพญากลาง จังหวัดลพบุรี และ (2ตำบลทับกวาง จังหวัดสระบุรี โดยมีวัตถุประสงค์ให้มีค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมและระยะทางรวมที่ประหยัดที่สุด โดยเริ่มต้นจากการสร้างแบบจำลองทางคณิตศาสตร์เพื่อหาคำตอบที่ดีที่สุด แต่ด้วยปัญหาของเวลาในการประมวลผลที่นานเมื่อเพิ่มจำนวนจุดของลูกค้าเพิ่มมากขึ้น จึงประยุกต์ใช้วิธีการทางฮิวริสติกโดยวิธี Saving Algorithm เพื่อหาคำตอบตั้งต้นของเส้นทางพบว่าได้คำตอบที่ใกล้เคียงคำตอบที่ดีที่สุด และพัฒนาคำตอบตั้งต้นเพื่อหาผลลัพธ์ที่ดีกว่าโดยวิธี Genetic Algorithm โดยผลลัพธ์เมื่อเปรียบเทียบกับวิธี Saving Algorithm ในกรณีโรงกำจัดซากไก่ตั้งที่ ตำบลลำพญากลาง พบว่า สามารถลดค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมและระยะทางรวมได้ 960,000 บาทต่อปี และ 51,000 กิโลเมตรต่อปี ตามลำดับ หรือ ร้อยละ 1.74 และ 2.05 ตามลำดับ และในกรณีโรงกำจัดซากไก่ตั้งที่ตำบลทับกวาง พบว่า สามารถลดค่าใช้จ่ายในการขนส่งแบบเหมาจ่ายรวมและระยะทางรวมได้ 1,860,000 บาทต่อปี และ 100,980 กม.ต่อปี ตามลำดับ หรือ ร้อยละ 3.39 และ 4.47 ตามลำดับ ซึ่งคำตอบเส้นทางที่ได้เป็นปัจจัยหนึ่งที่ส่งผลต่อการตัดสินใจของทางภาครัฐที่จะลงทุน

Chulalongkorn University

5.2 ข้อเสนอแนะ

การจัดรอบการขนส่งและลำดับเส้นทางจากงานวิจัยนี้ คิดจากสมมุติฐานที่มีการเดินรถเหมือนกันทุกวัน ทำให้ปริมาณที่ขนส่งบางจุดในแต่ละวันอาจไม่เต็มความจุของรถบรรทุก และมีบางรอบที่รถบรรทุกต้องไปรับวัตถุดิบจากหลายจุด (multiple pickup locations) ซึ่งในช่วงการดำเนินจริง หากต้องการให้รถบรรทุกรับวัตถุดิบเต็มคันในแต่ละจุด อาจปรับรอบการขนส่งให้แตกต่างกันในแต่ละวันได้ ซึ่งจะไม่ทำให้ค่าใช้จ่ายรวมสูงขึ้น เพราะเป็นการวางแผนที่คิดพร้อมกันหลายวัน (multiple period decision) แต่การหาเส้นทางที่ดีจะมีความซับซ้อนเพิ่มขึ้นและใช้เวลาในการจัดเส้นทางที่นานขึ้นมากด้วย

รายการอ้างอิง

[1] Keenan, P. Using a GIS as a DSS generator. Working paper MIS, pp.9-95 : Graduate School of Business, University College Dublin, 1997.

[2] เครือวัลย์ จำปาเงิน. การพัฒนาระบบสนับสนุนการตัดสินใจเพื่อจัดเส้นทางเดินรถสำหรับการขนส่งสินค้าเพื่อการบริโภคสู่ร้านค้าปลีกในสถานบริการน้ำมันในจังหวัดนนทบุรี. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาภูมิศาสตร์ คณะอักษรศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2547

[3] Moshe, D. Pierre, T. Split Delivery Routing. Naval Research Logistics, pp.383-402 : Centre de Recherche sur les Transports, Universite de Montreal, 1990.

[4] Ai, T. J. Particle Swarm Optimization for Generalized Vehicle Routing Problem. Doctoral Dissertation, Engineering and Technology Asian Institude of Technology, 2008.

[5] Clarke, G. Wright, J. V. Scheduling of Vehicle from a Central Depot to a Number of Delivery Point. pp. 568-581 : Operations Research, 1964.

[6] Holland, J. Adaptation in Natural and Artificial Systems. Ann Arbor : University of Michigan Press, 1975.

[7] ระพีพันธ์ ปิตาคะโส. วิธีการเมตาฮิวริสติกเพื่อแก้ไขปัญหาการวางแผนการผลิตและการจัดการโลจิสติกส์. กรุงเทพมหานคร : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2554

[8] อุบลรัตน์ เธียรธนาคม. การใช้วิธีเชิงฮิวริสติกส์เพื่อแก้ปัญหาการจัดเส้นทางยานพาหนะที่มีคลังสินค้าหลายแห่ง. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, 2551

[9] ศุภพัชร พวงแก้ว. วิธีการเชิงพันธุกรรมสำหรับปัญหาการจัดเส้นทางเดินรถเก็บขยะตามครัวเรือนโดยใช้คนงานในการจัดเก็บ. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2553

[10] ธฤต รัตนมณี. การพัฒนาฮิวริสติกบนพื้นฐานของอัลกอริทึมเชิงพันธุกรรมสำหรับการจัดเส้นทางรถขนส่งกรณีมีรถขนส่งหลายขนาดแบบหลายช่องบรรจุและหลายผลิตภัณฑ์ของการขนส่งน้ำมันเชื้อเพลิง. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2554

[11] พศวีร์ ตรีวิเศษ. รูปแบบการจัดการการขนส่งโดยการประยุกต์ใช้ขั้นตอนวิธีเชิงพันธุกรรม : กรณีศึกษาบริษัทขายวัสดุก่อสร้าง. วารสารมหาวิทยาลัยราชภัฏมหาสารคาม 5 (พฤษภาคม - สิงหาคม 2554) : 131-138.

[12] วันวิสา นฤมิตวงศ์. การจัดลำดับการผลิตที่มีหลายวัตถุประสงค์บนสายการประกอบผลิตภัณฑ์ผสมแบบสองด้านภายใต้ผลกระทบจากการเรียนรู้. วิทยานิพนธ์ปริญญามหาบัณฑิต, สาขาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2555

[13] K.C.Tan, L.H.Lee, Q.L.Zhu, K.Ou. Heristic methods for vehicle routing problem with time windows. pp. 281-295 : Artificial Intelligence in Engineering 15, 2001.

[14] ปฏิพัทธ์ หงษ์สุวรรณ. การศึกษาเพื่อเพิ่มประสิทธิภาพการจัดเส้นทางเดินรถขนส่งน้ำดื่มจังหวัดสมุทรสงคราม. วารสารการประชุมวิชาการข่ายงานวิศวกรรมอุตสาหการ ประจำปี พ.ศ. 2556, มหาวิทยาลัยศิลปากร

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ก
ตัวอย่าง การปรับปรุงเส้นทางตั้งต้นโดย Genetic Algorithm

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ตัวอย่าง การปรับปรุงเส้นทางตั้งต้นโดย Genetic Algorithm

จากตัวอย่างผลลัพธ์จากการจัดเส้นทางตั้งต้นไปรับสินค้าทั้งหมด 5 จุด โดย Saving Algorithm (กรณีโรงกำจัดซากตั้งที่ อ.ลำพญากลาง จ.ลพบุรี)

รอบการวิ่ง	เส้นทาง	น้ำหนัก (ตัน)	ระยะทาง (กม.ต่อวัน)	เวลา (นาที)	ค่าใช้จ่าย (บาทต่อวัน)
1	Depot -->ไทยโพลทรีกรุ๊ป (est.109) [3.19] --> GFN (est.201) [2.90] --> Depot	6.09	697.7	819.5	13,000
2	Depot --> ซีพีเอฟ (est.03) [5.22] --> ซีพีเอฟ (est.18) [1.74] -->Depot	6.96	550.1	689.3	11,300
3	Depot -->ไทยฟู้ดกรุ๊ป (est.119) [9.86] --> Depot	9.86	552	749.2	11,300
	รวม	22.91	1,799.8	2,258	35,600

รอบการวิ่ง	เส้นทาง
1	Depot --> 5 --> 4 --> depot
2	Depot --> 1 --> 2 --> depot
3	Depot --> 3 --> depot

คำตอบตั้งต้นจาก Saving Algorithm

เริ่มต้นกระบวนการด้วยการสร้างโครโมโซมแทนคำตอบจากคำตอบตั้งต้น เพื่อใช้เป็นโครโมโซมพ่อและแม่ตั้งต้นใน Genetic Algorithm โดยมีการจัดเซตของแต่ละยีนส์เป็น (ลำดับรอบการวิ่ง, ลำดับของลูกค้าภายในลำดับรอบการวิ่งนั้น) ดังนั้นจากคำตอบตั้งต้นจะสามารถสร้างโครโมโซมตั้งต้นได้นี้คือ

จุดของลูกค้า	1	2	3	4	5
ยีนส์	(2, 1)	(2, 2)	(3, 1)	(1, 2)	(1, 1)

โครโมโซมตั้งต้น = [(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]

จากนั้นทำการสลับยีนส์ภายในแต่ละลำดับรอบการวิ่งทั้งหมด และสลับยีนส์ระหว่างแต่ละรอบการวิ่ง โดยเงื่อนไขการสลับยีนส์ระหว่างแต่ละรอบการวิ่งมีวิธีการ คือ เลือกลำดับรอบการวิ่งที่ภายในมีจำนวนจุดของลูกค้า 2 และ 3 จุด แล้วนำแต่ละจุดของลูกค้าในลำดับรอบการวิ่งที่ได้เลือกนั้นไปทำการสลับกับจุดของลูกค้าอื่นๆที่ภายในลำดับรอบการวิ่งมีจำนวนจุดของลูกค้า 2 และ 3 จุดเหมือนกัน ทำการสลับตามเงื่อนไขจนครบทุกกรณี จะได้โครโมโซมตั้งต้นใหม่ แล้วทำการรวม

โครโมโซมตั้งต้นเดิมและใหม่รวมเป็น n โครโมโซม เพื่อนำมาเป็นประชากรตั้งต้น ในตัวอย่างนี้ จะสมมติสลับมาทั้งหมด 5 โครโมโซม เพื่อเป็นโครโมโซมตั้งต้น ดังนี้

โครโมโซม	เส้นทาง	ค่าใช้จ่าย (บาทต่อวัน)	ระยะทาง (กม.ต่อวัน)
1	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]	35,600	1,799.8
2	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]	35,600	1,799.8
3	[(2, 2) , (2, 1) , (3, 1) , (1, 2) , (1, 1)]	35,600	1,799.8
4	[(1, 1) , (2, 2) , (3, 1) , (1, 2) , (2, 1)]	38,700	1,924
5	[(2, 1) , (2, 2) , (1, 1) , (1, 2) , (3, 1)]	-	-

จากนั้นพิจารณาค่าความเหมาะสม (Fitness) ของประชากรตั้งต้น โดยมีการพิจารณาดังนี้

พิจารณาค่าความเหมาะสม (Fitness) ของประชากรตั้งต้น โดยพิจารณาจากสมการวัตถุประสงค์โดยใช้ค่าใช้จ่ายแบบเหมารวมของทุกรอบการวิ่ง โดยกำหนดให้ค่าความเหมาะสมของโครโมโซมที่มีค่าใช้จ่ายน้อยที่สุด มีค่ามากที่สุด ตามลำดับ

โครโมโซม	เส้นทาง	ค่าใช้จ่าย (บาทต่อวัน)	ระยะทาง (กม.ต่อวัน)	ค่าความเหมาะสม
1	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]	35,600	1,799.8	3
2	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]	35,600	1,799.8	3
3	[(2, 2) , (2, 1) , (3, 1) , (1, 2) , (1, 1)]	35,600	1,799.8	3
4	[(1, 1) , (2, 2) , (3, 1) , (1, 2) , (2, 1)]	38,700	1,924	2
5	[(2, 1) , (2, 2) , (1, 1) , (1, 2) , (3, 1)]	-	-	1

ทำการคัดเลือกโครโมโซมโดยการใช้การสร้าง Roulette Wheel โดยหาค่าผลรวมของ Fitness ของโครโมโซมทั้งหมด แล้วจึงหาค่าความน่าจะเป็นในการถูกคัดเลือก (Probability of Selection : p_i) และความน่าจะเป็นในการถูกคัดเลือกสะสม (Cumulative Probability of Selection : q_i) ของโครโมโซมแต่ละตัว

โครโมโซม	ค่าความเหมาะสม	p_i	q_i
1	3	0.25	0.25
2	3	0.25	0.5
3	3	0.25	0.75
4	2	0.166	0.916
5	1	0.083	1
	12		

ทำการคัดเลือกโครโมโซมด้วยวิธีการ Binary Tournament Selection โดยการสร้างตัวเลขสุ่ม (r) ที่มีค่าระหว่าง 0 ถึง 1 โดยเมื่อสุ่มโครโมโซมสองตัวออกมาแล้วให้นำมาพิจารณาค่าความเหมาะสมของโครโมโซมแต่ละตัว ถ้าโครโมโซมใดมีค่าความเหมาะสมมากกว่าให้เลือกโครโมโซมนั้น แต่ถ้าทั้งสองโครโมโซมมีค่าความเหมาะสมเท่ากัน ให้พิจารณาจากค่า ค่าใช้จ่ายการขนส่งแบบเหมาจ่ายรวม ถ้าโครโมโซมใดมีค่า ค่าใช้จ่ายการขนส่งแบบเหมาจ่ายน้อยกว่าให้เลือกโครโมโซมนั้น แต่ถ้าทั้งสองโครโมโซมมีค่า ค่าใช้จ่ายการขนส่งแบบเหมาจ่ายเท่ากัน ให้พิจารณาจาก ค่าระยะทางรวม ถ้าโครโมโซมใดมีค่า ระยะทางรวม น้อยกว่าให้เลือกโครโมโซมนั้น แต่ถ้าทั้งสองโครโมโซมมีค่าระยะทางรวมเท่ากันอีก ให้สุ่มเลือกโครโมโซม 1 โครโมโซม แล้วจึงนำเข้าสู่ Mating Pool

ลำดับ	ประชากรที่ 1				ประชากรที่ 2				ลำดับโครโมโซมที่คัดเลือก
	r_1	$r_1 < q_i$	โครโมโซม	ค่าความเหมาะสม	r_2	$r_2 < q_i$	โครโมโซม	ค่าความเหมาะสม	
1	0.7894	0.916	4	2	0.9843	1	5	1	4
2	0.5678	0.75	3	3	0.8457	0.916	4	2	3
3	0.2245	0.25	1	3	0.3246	0.5	2	3	1
4	0.2938	0.5	2	3	0.876	0.916	4	2	2
5	0.1123	0.25	1	3	0.987	1	5	1	1

จากการคัดเลือกโครโมโซมด้วยวิธี Binary Tournament Selection โครโมโซมที่ได้คัดเลือกเข้าสู่ Mating Pool คือ โครโมโซมที่ 4 3 1 2 และ 1 ตามลำดับ

การ Crossover

เป็นการนำโครโมโซมที่อยู่ใน Mating Pool ที่ได้จากการคัดเลือกด้วยวิธี Binary Tournament Selection มาทำการจับคู่กันเพื่อแลกเปลี่ยนส่วนของยีนส์ซึ่งกันและกัน ทำให้เกิดโครโมโซมตัวใหม่ ส่วนโครโมโซมที่ไม่ได้ถูกทำการจับคู่เพื่อครอสโอเวอร์ก็ยังคงสภาพเดิมอยู่ใน Mating Pool โดยมีขั้นตอน ดังนี้

1. หาจำนวนโครโมโซมที่ถูกนำมาจับคู่ในการครอสโอเวอร์ ในตัวอย่างนี้จะกำหนดให้ความน่าจะเป็นในการครอสโอเวอร์เท่ากับ 0.7 ดังนั้น โครโมโซมที่ถูกครอสโอเวอร์จะมีทั้งหมด 0.7 x 5 = 3.5 หรือ 4 โครโมโซม
2. ทำการสุ่มค่าความน่าจะเป็นที่มีค่าอยู่ระหว่าง 0-1 ให้กับโครโมโซม ถ้าค่าความน่าจะเป็นที่สุ่มได้มีค่าน้อยกว่า 0.7 โครโมโซมนั้นจะถูกนำไปจับคู่ แต่ถ้าทำการสุ่มความน่าจะเป็นแล้วไม่มีโครโมโซมใดเลยที่มีค่าน้อยกว่า 0.7 ให้ทำการสุ่มใหม่เรื่อยๆจนกว่าจะได้โครโมโซม
3. ถ้าจำนวนโครโมโซมที่ถูกนำมาจับคู่มีค่าเป็นจำนวนคี่ สามารถพิจารณาได้ดังนี้

- ถ้ามีโครโมโซมที่ถูกนำมาจับคู่เพียง 1 ตัว ให้ทำการสุ่มโครโมโซมที่เหลือภายใน Mating Pool เพิ่มเข้าไปอีก 1 ตัว

- ถ้ามีโครโมโซมที่ถูกนำมาจับคู่เป็นจำนวนคี่ที่มีจำนวนมากว่า 1 ตัว แต่น้อยกว่าจำนวนโครโมโซมทั้งหมดภายใน Mating Pool ให้ทำการสุ่มโครโมโซมภายใน Mating Pool ที่เหลือเพิ่มอีก 1 ตัว
- ถ้ามีโครโมโซมที่ถูกนำมาจับคู่มีจำนวนเท่ากับโครโมโซมทั้งหมดภายใน Mating Pool ที่เป็นจำนวนคี่ ให้ทำการสุ่มตัดโครโมโซมทิ้ง 1 ตัว

4. ให้ทำการจับคู่ของโครโมโซมที่ได้สุ่มเลือกไว้ โดยเรียงจับคู่ตามลำดับ

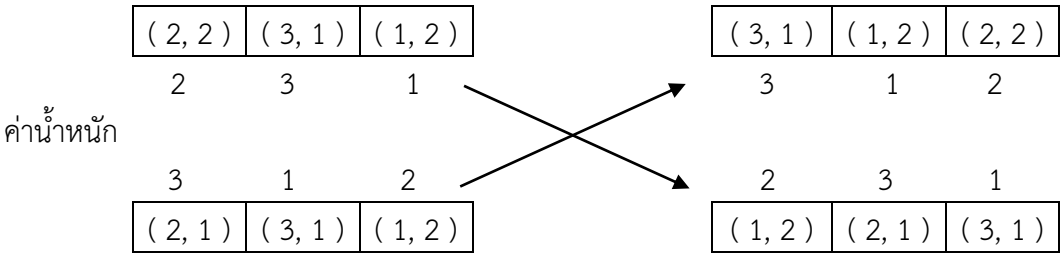
โครโมโซมได้ที่เข้าสู่ Mating Pool ที่ทำการ Crossover

โครโมโซม	**โครโมโซมที่คัดเลือก**	**เส้นทาง**	r_i	r_i **<0.7**
1	4	[(1, 1) , (2, 2) , (3, 1) , (1, 2) , (2, 1)]	0.4567	คัดเลือก
2	3	[(2, 2) , (2, 1) , (3, 1) , (1, 2) , (1, 1)]	0.6789	คัดเลือก
3	1	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]	0.9342	-
4	2	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]	0.2345	คัดเลือก
5	1	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]	0.1234	คัดเลือก

ได้โครโมโซมที่จะนำไปทำการ Crossover คือ โครโมโซมที่ 1 2 4 และ 5 ซึ่งจับคู่ได้เป็น 1 , 2 และ 4 , 5 โดยทำหน้าที่เป็นโครโมโซมรุ่นพ่อแม่ (Parent) โดยจะถูกนำไปแลกเปลี่ยนยีนส์ภายในโครโมโซมซึ่งกันและกัน แล้วได้ออกมาเป็นโครโมโซมรุ่นลูก (Offspring) โดยจะใช้วิธีการครอสโอเวอร์แบบ Weighted Mapping Crossover (WMX) มีขั้นตอน ดังนี้

- สุ่มตำแหน่งที่จะการครอสโอเวอร์ โดยต้องมียีนส์ภายในช่วงอย่างน้อย 2 ตัว และตำแหน่งที่จะทำการสุ่มภายในโครโมโซมพ่อและแม่ต้องเป็นตำแหน่งเดียวกันเท่านั้น จากตัวอย่าง มียีนส์ภายในช่วงทั้งหมด 3 ตัว และตำแหน่งที่ทำการครอสโอเวอร์ระหว่างโครโมโซมพ่อและแม่ คือ ตำแหน่งลูกค้าที่ 2 3 และ 4
- ค่าน้ำหนักที่กำหนดลงในแต่ละตำแหน่งของยีนส์นั้นได้มาจากการสุ่มตามลำดับจำนวนยีนส์ทั้งหมดที่ทำการครอสโอเวอร์ และทำการสลับค่าน้ำหนักที่กำหนดระหว่างยีนส์ของพ่อและแม่ จึงทำให้ลำดับยีนส์ภายในโครโมโซมของพ่อและแม่เปลี่ยนแปลง

Parent 1 = [(1, 1) | (2, 2) , (3, 1) , (1, 2) | (2, 1)]
Parent 2 = [(2, 2) | (2, 1) , (3, 1) , (1, 2) | (1, 1)]



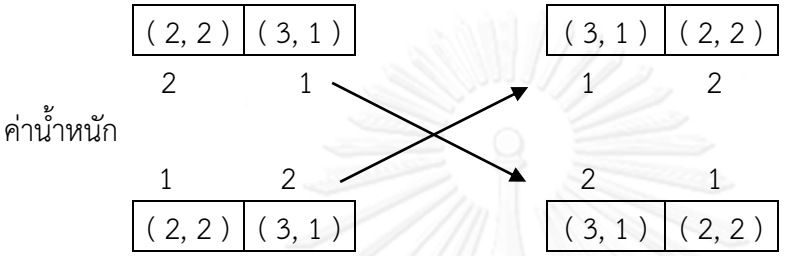
ดังนั้น จะได้โครโมโซมในรุ่นลูก (Offspring) คือ

Offspring 1 = [(1, 1) | (3, 1) , (1, 2) , (2, 2) | (2, 1)]

Offspring 2 = [(2, 2) | (1, 2) , (2, 1) , (3, 1) | (1, 1)]

Parent 4 = [(2, 1) | (2, 2) , (3, 1) | (1, 1) , (1, 2)]

Parent 5 = [(2, 1) | (2, 2) , (3, 1) | (1, 2) , (1, 1)]



ดังนั้น จะได้โครโมโซมในรุ่นลูก (Offspring) คือ

Offspring 4 = [(2, 1) | (3, 1) (2, 2) | (1, 1) , (1, 2)]

Offspring 5 = [(2, 1) | (3, 1) (2, 2) | (1, 2) , (1, 1)]

โครโมโซมที่ทำการ Crossover แล้ว

โครโมโซม	**เส้นทาง**
1	[(1, 1) , (3, 1) , (1, 2) , (2, 2) , (2, 1)]
2	[(2, 2) , (1, 2) , (2, 1) , (3, 1) , (1, 1)]
3	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]
4	[(2, 1) , (3, 1) (2, 2) , (1, 1) , (1, 2)]
5	[(2, 1) , (3, 1) (2, 2) , (1, 2) , (1, 1)]

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การ Mutation

เป็นการนำยีนส์ภายในโครโมโซมเดียวกันมาทำการสลับตำแหน่ง โดยการมิวเตชั่นจะเป็นกระบวนการที่ทำหลังจากผ่านการครอสโอเวอร์มาแล้ว โดยการมิวเตชั่นมีขั้นตอน ดังนี้

1. หาจำนวนโครโมโซมที่ถูกนำมาจับคู่ในการมิวเตชั่น ในตัวอย่างนี้จะกำหนดให้ความน่าจะเป็นในการมิวเตชั่นเท่ากับ 0.3 ดังนั้น โครโมโซมที่ถูกมิวเตชั่นจะมีทั้งหมด 0.3 x 5 = 1.5 หรือ 2 โครโมโซม

2. ทำการสุ่มค่าความน่าจะเป็นที่มีค่าอยู่ระหว่าง 0-1 ให้กับโครโมโซม ถ้าค่าความน่าจะเป็นที่สุ่มได้มีค่าน้อยกว่า 0.3 โครโมโซมนั้นจะถูกนำไปจับคู่ แต่ถ้าทำการสุ่มความน่าจะเป็นแล้วไม่มีโครโมโซมใดเลยที่มีค่าน้อยกว่า 0.3 ให้ทำการสุ่มใหม่เรื่อยๆจนกว่าจะได้โครโมโซม

โครโมโซมที่จะทำการ Mutation

โครโมโซม	**เส้นทาง**	r_i	r_i **<0.3**
1	[(1, 1) , (3, 1) , (1, 2) , (2, 2) , (2, 1)]	0.4678	-
2	[(2, 2) , (1, 2) , (2, 1) , (3, 1) , (1, 1)]	0.6876	-
3	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]	0.2015	คัดเลือก
4	[(2, 1) , (3, 1) , (2, 2) , (1, 1) , (1, 2)]	0.1078	คัดเลือก
5	[(2, 1) , (3, 1) , (2, 2) , (1, 2) , (1, 1)]	0.7988	-

ได้โครโมโซมที่จะนำไปทำการ Mutation คือ โครโมโซมที่ 3 และ 4 โดยทำหน้าที่เป็นโครโมโซมรุ่นพ่อแม่ (Parent) โดยจะถูกนำไปแลกเปลี่ยนยีนส์ภายในโครโมโซมตัวเอง แล้วได้ออกมาเป็นโครโมโซมรุ่นลูก (Offspring) โดยจะใช้วิธีการมิวเตชั่นแบบ Reciprocal Exchange Mutation ซึ่งจะทำการสุ่มตำแหน่งยีนส์ภายในโครโมโซม 2 ตำแหน่ง แล้วนำมาสลับตำแหน่งกัน จากตัวอย่างโครโมโซมที่ 3 ได้ทำการสุ่มได้ ตำแหน่งยีนส์ที่ 2 และ 5 แล้วทำการสลับตำแหน่งของยีนส์ มีขั้นตอนดังนี้

Parent 3 = [(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]
Offspring 3 = [(2, 1) , (1, 1) , (3, 1) , (1, 2) , (2, 2)]

Parent 4 = [(2, 1) , (3, 1) , (2, 2) , (1, 1) , (1, 2)]
Offspring 4 = [(1, 1) , (3, 1) , (2, 2) , (2, 1), (1, 2)]

โครโมโซมที่ทำการ Mutation แล้ว

โครโมโซม	เส้นทาง
1	[(1, 1) , (3, 1) , (1, 2) , (2, 2) , (2, 1)]
2	[(2, 2) , (1, 2) , (2, 1) , (3, 1) , (1, 1)]
3	[(2, 1) , (1, 1) , (3, 1) , (1, 2) , (2, 2)]
4	[(1, 1) , (3, 1) , (2, 2) , (2, 1), (1, 2)]
5	[(2, 1) , (3, 1) , (2, 2) , (1, 2) , (1, 1)]

การเก็บค่าที่ดีที่สุด

หลังจากผ่านกระบวนการต่าง ทั้งการครอสโอเวอร์ และการมิวเตชั่น อาจทำให้เกิดคำตอบที่แย่กว่าก่อนการผ่านกระบวนการปรับปรุง ดังนั้นเทคนิคการเก็บค่าที่ดีที่สุดจึงต้องทำการเก็บค่าโครโมโซมทั้งรุ่นพ่อแม่และรุ่นลูกมารวมกัน แล้วจึงหาค่าโครโมโซมที่ดีที่สุดไว้เพื่อป้องกันไม่ให้คำตอบที่ดีสูญหายไปได้

โครโมโซมรุ่นพ่อแม่และโครโมโซมรุ่นลูก

รุ่นโครโมโซม	โครโมโซม	เส้นทาง
โครโมโซมรุ่นพ่อแม่ (Parent)	1	[(1, 1) , (2, 2) , (3, 1) , (1, 2) , (2, 1)]
	2	[(2, 2) , (2, 1) , (3, 1) , (1, 2) , (1, 1)]
	3	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]
	4	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]
	5	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]
โครโมโซมรุ่นลูก (Offspring)	6	[(1, 1) , (3, 1) , (1, 2) , (2, 2) , (2, 1)]
	7	[(2, 2) , (1, 2) , (2, 1) , (3, 1) , (1, 1)]
	8	[(2, 1) , (1, 1) , (3, 1) , (1, 2) , (2, 2)]
	9	[(1, 1) , (3, 1) , (2, 2) , (2, 1), (1, 2)]
	10	[(2, 1) , (3, 1) , (2, 2) , (1, 2) , (1, 1)]

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ทำการตัดโครโมโซมที่ซ้ำกันออก จากนั้นคำนวณหา ค่าใช้จ่ายรวม และ ระยะทางรวม เพื่อพิจารณาค่าความเหมาะสม

โครโมโซม	เส้นทาง	ค่าใช้จ่าย (บาทต่อวัน)	ระยะทาง (กม.ต่อวัน)	ค่าความเหมาะสม
1	[(1, 1), (2, 2), (3, 1), (1, 2), (2, 1)]	38,700	1,924	2
2	[(2, 2), (2, 1), (3, 1), (1, 2), (1, 1)]	35,600	1,799.8	3
3	[(2, 1), (2, 2), (3, 1), (1, 2), (1, 1)]	35,600	1,799.8	3
4	[(2, 1), (2, 2), (3, 1), (1, 1), (1, 2)]	35,600	1,799.8	3
5	[(1, 1), (3, 1), (1, 2), (2, 2), (2, 1)]	-	-	1
6	[(2, 2), (1, 2), (2, 1), (3, 1), (1, 1)]	-	-	1
7	[(2, 1), (1, 1), (3, 1), (1, 2), (2, 2)]	38,700	1,927	2
8	[(1, 1), (3, 1), (2, 2), (2, 1), (1, 2)]	-	-	1
9	[(2, 1), (3, 1), (2, 2), (1, 2), (1, 1)]	-	-	1

ทำการจัดลำดับใหม่โดยพิจารณาจากค่าความเหมาะสม จากมากไปน้อย ถ้าค่าความเหมาะสม เท่ากันให้พิจารณาจากค่าใช้จ่ายรวมจากน้อยไปมาก และถ้าค่าใช้จ่ายรวมเท่ากันให้พิจารณาจากค่าระยะทางรวมจากน้อยไปมาก ตามลำดับ

โครโมโซม	เส้นทาง	ค่าใช้จ่าย (บาทต่อวัน)	ระยะทาง (กม.ต่อวัน)	ค่าความเหมาะสม
2	[(2, 2), (2, 1), (3, 1), (1, 2), (1, 1)]	35,600	1,799.8	3
3	[(2, 1), (2, 2), (3, 1), (1, 2), (1, 1)]	35,600	1,799.8	3
4	[(2, 1), (2, 2), (3, 1), (1, 1), (1, 2)]	35,600	1,799.8	3
1	[(1, 1), (2, 2), (3, 1), (1, 2), (2, 1)]	38,700	1,924	2
7	[(2, 1), (1, 1), (3, 1), (1, 2), (2, 2)]	38,700	1,927	2
5	[(1, 1), (3, 1), (1, 2), (2, 2), (2, 1)]	-	-	1
6	[(2, 2), (1, 2), (2, 1), (3, 1), (1, 1)]	-	-	1
8	[(1, 1), (3, 1), (2, 2), (2, 1), (1, 2)]	-	-	1
9	[(2, 1), (3, 1), (2, 2), (1, 2), (1, 1)]	-	-	1

ทำการคัดเลือกโครโมโซมที่อยู่ใน 5 อันดับแรก (ให้เท่ากับจำนวนโครโมโซมตั้งต้น) เพื่อทำการเก็บค่าที่ดีที่สุดไว้ดำเนินการใน Genetic Algorithm ในรอบถัดไป

โครโมโซมที่ดีที่สุดในรอบแรกและเป็นโครโมโซมตั้งต้นในรอบถัดไป

โครโมโซม	เส้นทาง
2	[(2, 2), (2, 1), (3, 1), (1, 2), (1, 1)]
3	[(2, 1), (2, 2), (3, 1), (1, 2), (1, 1)]
4	[(2, 1), (2, 2), (3, 1), (1, 1), (1, 2)]
1	[(1, 1), (2, 2), (3, 1), (1, 2), (2, 1)]
7	[(2, 1), (1, 1), (3, 1), (1, 2), (2, 2)]

การหาคำตอบโดย Genetic Algorithm ในรอบที่ 2

หลังจากได้โครโมโซมที่ดีที่สุดในรอบที่ 1 แล้วจะนำโครโมโซมที่ได้นั้นมาเป็นคำตอบตั้งต้นในกระบวนการในรอบที่ 2 โดยการแก้ปัญหาในรอบที่ 2 มีลักษณะที่เหมือนกับในรอบที่ 1 โดยเริ่มต้นพิจารณาจากค่าความเหมาะสม

โครโมโซม	เส้นทาง	ค่าใช้จ่าย (บาทต่อวัน)	ระยะทาง (กม.ต่อวัน)	ค่าความเหมาะสม
1	[(2, 2), (2, 1), (3, 1), (1, 2), (1, 1)]	35,600	1,799.8	2
2	[(2, 1), (2, 2), (3, 1), (1, 2), (1, 1)]	35,600	1,799.8	2
3	[(2, 1), (2, 2), (3, 1), (1, 1), (1, 2)]	35,600	1,799.8	2
4	[(1, 1), (2, 2), (3, 1), (1, 2), (2, 1)]	38,700	1,924	1
5	[(2, 1), (1, 1), (3, 1), (1, 2), (2, 2)]	38,700	1,927	1

ทำการคัดเลือกโครโมโซมโดยการใช้การสร้าง Roulette Wheel โดยหาค่าผลรวมของ Fitness ของโครโมโซมทั้งหมด แล้วจึงหาค่าความน่าจะเป็นในการถูกคัดเลือก (Probability of Selection : p_i) และความน่าจะเป็นในการถูกคัดเลือกสะสม (Cumulative Probability of Selection : q_i) ของโครโมโซมแต่ละตัว

โครโมโซม	ค่าความเหมาะสม	p_i	q_i
1	2	0.25	0.25
2	2	0.25	0.5
3	2	0.25	0.75
4	1	0.125	0.875
5	1	0.125	1
	8		

ทำการคัดเลือกโครโมโซมด้วยวิธีการ Binary Tournament Selection โดยการสร้างตัวเลขสุ่ม (r) ที่มีค่าระหว่าง 0 ถึง 1 ออกมาสองค่า โดยเมื่อสุ่มโครโมโซมสองตัวออกมาแล้วให้นำมาพิจารณาค่าความเหมาะสมของโครโมโซมแต่ละตัว ถ้าโครโมโซมใดมีค่าความเหมาะสมมากกว่าให้เลือกโครโมโซมนั้น แต่ถ้าทั้งสองโครโมโซมมีค่าความเหมาะสมเท่ากัน ให้พิจารณาจากค่า ค่าใช้จ่ายการขนส่งแบบเหมาจ่ายรวม ถ้าโครโมโซมใดมีค่า ค่าใช้จ่ายการขนส่งแบบเหมาจ่ายน้อยกว่าให้เลือกโครโมโซมนั้น แต่ถ้าทั้งสองโครโมโซมมีค่า ค่าใช้จ่ายการขนส่งแบบเหมาจ่ายเท่ากัน ให้พิจารณาจากค่าระยะทางรวม ถ้าโครโมโซมใดมีค่า ระยะทางรวม น้อยกว่าให้เลือกโครโมโซมนั้น แต่ถ้าทั้งสองโครโมโซมมีค่าระยะทางรวมเท่ากันอีก ให้สุ่มเลือกโครโมโซม 1 โครโมโซม แล้วจึงนำเข้าสู่ Mating Pool

ลำดับ	ประชากรที่ 1				ประชากรที่ 2				ลำดับโครโมโซมที่คัดเลือก
	r_1	$r_1 < q_i$	โครโมโซม	ค่าความเหมาะสม	r_2	$r_2 < q_i$	โครโมโซม	ค่าความเหมาะสม	
1	0.5678	0.75	3	2	0.3987	0.5	2	2	3
2	0.678	0.75	3	2	0.8764	1	5	1	3
3	0.9543	1	5	1	0.2864	0.5	2	2	2
4	0.2865	0.5	2	2	0.9753	1	5	1	2
5	0.7543	0.875	4	1	0.1276	0.25	1	2	1

จากการคัดเลือกโครโมโซมด้วยวิธี Binary Tournament Selection โครโมโซมที่ได้คัดเลือกเข้าสู่ Mating Pool คือ สตริงคำตอบที่ 3 3 2 2 และ 1 ตามลำดับ

การ Crossover

เป็นการนำโครโมโซมที่อยู่ใน Mating Pool ที่ได้จากการคัดเลือกด้วยวิธี Binary Tournament Selection มาทำการจับคู่กันเพื่อแลกเปลี่ยนส่วนของยีนส์ซึ่งกันและกัน ทำให้เกิดโครโมโซมตัวใหม่ ส่วนโครโมโซมที่ไม่ได้ถูกทำการจับคู่เพื่อครอสโอเวอร์ก็ยังคงสภาพเดิมอยู่ใน Mating Pool โดยมีขั้นตอน ดังนี้

1. หาจำนวนโครโมโซมที่ถูกนำมาจับคู่ในการครอสโอเวอร์ ในตัวอย่างนี้จะกำหนดให้ความน่าจะเป็นในการครอสโอเวอร์เท่ากับ 0.7 ดังนั้น โครโมโซมที่ถูกครอสโอเวอร์จะมีทั้งหมด 0.7 x 5 = 3.5 หรือ 4 โครโมโซม
2. ทำการสุ่มค่าความน่าจะเป็นที่มีค่าอยู่ระหว่าง 0-1 ให้กับโครโมโซม ถ้าค่าความน่าจะเป็นที่สุ่มได้มีค่าน้อยกว่า 0.7 โครโมโซมนั้นจะถูกนำไปจับคู่ แต่ถ้าทำการสุ่มความน่าจะเป็นแล้วไม่มีโครโมโซมใดเลยที่มีค่าน้อยกว่า 0.7 ให้ทำการสุ่มใหม่เรื่อยๆจนกว่าจะได้โครโมโซม
3. ถ้าจำนวนโครโมโซมที่ถูกนำมาจับคู่มีค่าเป็นจำนวนคี่ สามารถพิจารณาได้ดังนี้

- ถ้ามีโครโมโซมที่ถูกนำมาจับคู่เพียง 1 ตัว ให้ทำการสุ่มโครโมโซมที่เหลือภายใน Mating Pool เพิ่มเข้าไปอีก 1 ตัว

- ถ้ามีโครโมโซมที่ถูกนำมาจับคู่เป็นจำนวนคี่ที่มีจำนวนมากว่า 1 ตัว แต่น้อยกว่าจำนวนโครโมโซมทั้งหมดภายใน Mating Pool ให้ทำการสุ่มโครโมโซมภายใน Mating Pool ที่เหลือเพิ่มอีก 1 ตัว
- ถ้ามีโครโมโซมที่ถูกนำมาจับคู่มีจำนวนเท่ากับโครโมโซมทั้งหมดภายใน Mating Pool ที่เป็นจำนวนคี่ ให้ทำการสุ่มตัดโครโมโซมทิ้ง 1 ตัว

4. ให้ทำการจับคู่ของโครโมโซมที่ได้สุ่มเลือกไว้ โดยเรียงจับคู่ตามลำดับ

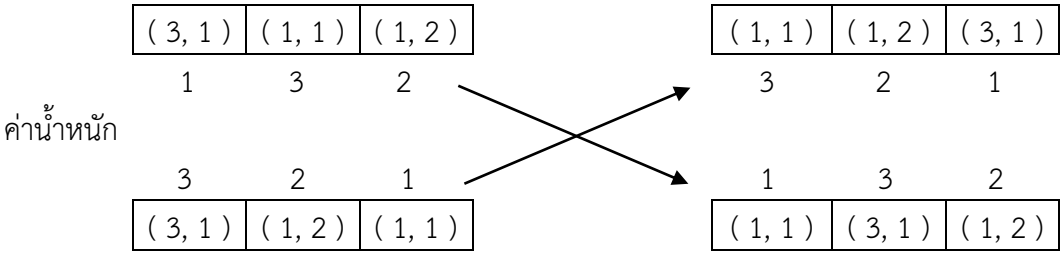
โครโมโซมได้ที่เข้าสู่ Mating Pool ที่ทำการ Crossover

โครโมโซม	**โครโมโซมที่คัดเลือก**	**เส้นทาง**	r_i	r_i **<0.7**
1	3	[(2, 1), (2, 2), (3, 1), (1, 1), (1, 2)]	0.8753	-
2	3	[(2, 1), (2, 2), (3, 1), (1, 1), (1, 2)]	0.2543	คัดเลือก
3	2	[(2, 1), (2, 2), (3, 1), (1, 2), (1, 1)]	0.6793	คัดเลือก
4	2	[(2, 1), (2, 2), (3, 1), (1, 2), (1, 1)]	0.5421	คัดเลือก
5	1	[(2, 2), (2, 1), (3, 1), (1, 2), (1, 1)]	0.2354	คัดเลือก

ได้โครโมโซมที่จะนำไปทำการ Crossover คือ โครโมโซมที่ 2 3 4 และ 5 ซึ่งจับคู่ได้เป็น 2 , 3 และ 4 , 5 โดยทำหน้าที่เป็นโครโมโซมรุ่นพ่อแม่ (Parent) โดยจะถูกนำไปแลกเปลี่ยนยีนส์ภายในโครโมโซมซึ่งกันและกัน แล้วได้ออกมาเป็นโครโมโซมรุ่นลูก (Offspring) โดยจะใช้วิธีการครอสโอเวอร์แบบ Weighted Mapping Crossover (WMX) มีขั้นตอน ดังนี้

- สุ่มตำแหน่งที่จะการครอสโอเวอร์ โดยต้องมียีนส์ภายในช่วงอย่างน้อย 2 ตัว และตำแหน่งที่จะทำการสุ่มภายในโครโมโซมพ่อและแม่ต้องเป็นตำแหน่งเดียวกันเท่านั้น
- ค่าน้ำหนักที่กำหนดลงในแต่ละตำแหน่งของยีนส์นั้นได้มาจากการสุ่มตามลำดับจำนวนยีนส์ทั้งหมดที่ทำการครอสโอเวอร์ และทำการสลับค่าน้ำหนักที่กำหนดระหว่างยีนส์ของพ่อและแม่ จึงทำให้ลำดับยีนส์ภายในโครโมโซมของพ่อและแม่เปลี่ยนแปลง

Parent 2 = [(2, 1), (2, 2) | (3, 1), (1, 1), (1, 2) |]
Parent 3 = [(2, 1), (2, 2) | (3, 1), (1, 2), (1, 1) |]



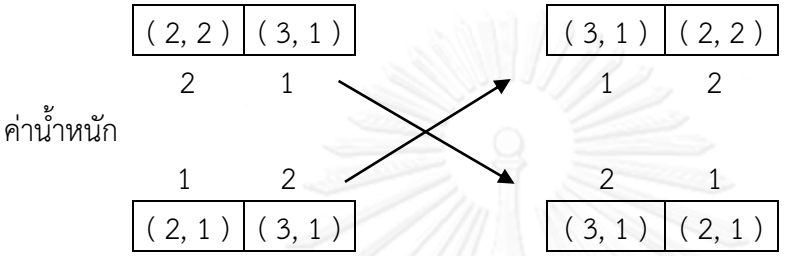
ดังนั้น จะได้โครโมโซมในรุ่นลูก (Offspring) คือ

Offspring 2 = [(2, 1) , (2, 2) | (1, 1) , (1, 2) , (3, 1) |]

Offspring 3 = [(2, 1) , (2, 2) | (1, 1) , (3, 1) , (1, 2) |]

Parent 4 = [(2, 1) | (2, 2) , (3, 1) | (1, 2) , (1, 1)]

Parent 5 = [(2, 2) | (2, 1) , (3, 1) | (1, 2) , (1, 1)]



ดังนั้น จะได้โครโมโซมในรุ่นลูก (Offspring) คือ

Offspring 4 = [(2, 1) | (3, 1) , (2, 2) | (1, 2) , (1, 1)]

Offspring 5 = [(2, 2) | (3, 1) , (2, 1) | (1, 2) , (1, 1)]

โครโมโซมที่ทำการ Crossover แล้ว

โครโมโซม	เส้นทาง
1	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]
2	[(2, 1) , (2, 2) , (1, 1) , (1, 2) , (3, 1)]
3	[(2, 1) , (2, 2) , (1, 1) , (3, 1) , (1, 2)]
4	[(2, 1) , (3, 1) , (2, 2) , (1, 2) , (1, 1)]
5	[(2, 2) , (3, 1) , (2, 1) , (1, 2) , (1, 1)]

การ Mutation

เป็นการนำยีนส์ภายในโครโมโซมเดียวกันมาทำการสลับตำแหน่ง โดยการมิวเตชั่นจะเป็นกระบวนการที่ทำหลังจากผ่านการครอสโอเวอร์มาแล้ว โดยการมิวเตชั่นมีขั้นตอน ดังนี้

1. หาจำนวนโครโมโซมที่ถูกนำมาจับคู่ในการมิวเตชั่น ในตัวอย่างนี้จะกำหนดให้ความน่าจะเป็นในการมิวเตชั่นเท่ากับ 0.3 ดังนั้น โครโมโซมที่ถูกมิวเตชั่นจะมีทั้งหมด 0.3 x 5 = 1.5 หรือ 2 โครโมโซม
2. ทำการสุ่มค่าความน่าจะเป็นที่มีค่าอยู่ระหว่าง 0-1 ให้กับโครโมโซม ถ้าค่าความน่าจะเป็นที่สุ่มได้มีค่าน้อยกว่า 0.3 โครโมโซมนั้นจะถูกนำไปจับคู่ แต่ถ้าทำการสุ่มความน่าจะเป็นแล้วไม่มีโครโมโซมใดเลยที่มีค่าน้อยกว่า 0.3 ให้ทำการสุ่มใหม่เรื่อยๆจนกว่าจะได้โครโมโซม

โครโมโซมที่จะทำการ Mutation

โครโมโซม	เส้นทาง	r_i	r_i <0.3
1	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]	0.7532	-
2	[(2, 1) , (2, 2) , (1, 1) , (1, 2) , (3, 1)]	0.1364	คัดเลือก
3	[(2, 1) , (2, 2) , (1, 1) , (3, 1) , (1, 2)]	0.2531	คัดเลือก
4	[(2, 1) , (3, 1) , (2, 2) , (1, 2) , (1, 1)]	0.8743	-
5	[(2, 2) , (3, 1) , (2, 1) , (1, 2) , (1, 1)]	0.5431	-

ได้โครโมโซมที่จะนำไปทำการ Mutation คือ โครโมโซมที่ 3 และ 4 โดยทำหน้าที่เป็นโครโมโซมรุ่นพ่อแม่ (Parent) โดยจะถูกนำไปแลกเปลี่ยนยีนส์ภายในโครโมโซมตัวเอง แล้วได้ออกมาเป็นโครโมโซมรุ่นลูก (Offspring) โดยจะใช้วิธีการมิวเตชั่นแบบ Reciprocal Exchange Mutation ซึ่งจะทำการสุ่มตำแหน่งยีนส์ภายในโครโมโซม 2 ตำแหน่ง แล้วนำมาสลับตำแหน่งกัน มีขั้นตอน ดังนี้

Parent 2 = [(2, 1) , (2, 2) , <u>(1, 1)</u> , (1, 2) , <u>(3, 1)</u>]
Offspring 2 = [(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]

Parent 3 [(2, 1) , (2, 2) , <u>(1, 1)</u> , <u>(3, 1)</u> , (1, 2)]
Offspring 3 = [(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]

โครโมโซมที่ทำการ Mutation แล้ว

โครโมโซม	เส้นทาง
1	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]
2	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]
3	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]
4	[(2, 1) , (3, 1) , (2, 2) , (1, 2) , (1, 1)]
5	[(2, 2) , (3, 1) , (2, 1) , (1, 2) , (1, 1)]

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

การเก็บค่าที่ดีที่สุด

หลังจากผ่านกระบวนการต่าง ทั้งการครอสโอเวอร์ และการมิวเตชั่น อาจทำให้เกิดคำตอบที่แย่กว่าก่อนการผ่านกระบวนการปรับปรุง ดังนั้นเทคนิคการเก็บค่าที่ดีที่สุดจึงต้องทำการเก็บค่าโครโมโซมทั้งรุ่นพ่อแม่และรุ่นลูกมารวมกัน แล้วจึงหาค่าโครโมโซมที่ดีที่สุดไว้เพื่อป้องกันไม่ให้คำตอบที่ดีสูญหายไปได้

โครโมโซมรุ่นพ่อแม่และโครโมโซมรุ่นลูก

รุ่นโครโมโซม	โครโมโซม	เส้นทาง
โครโมโซมรุ่นพ่อแม่ (Parent)	1	[(2, 2) , (2, 1) , (3, 1) , (1, 2) , (1, 1)]
	2	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]
	3	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]
	4	[(1, 1) , (2, 2) , (3, 1) , (1, 2) , (2, 1)]
	5	[(2, 1) , (1, 1) , (3, 1) , (1, 2) , (2, 2)]
โครโมโซมรุ่นลูก (Offspring)	6	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]
	7	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]
	8	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]
	9	[(2, 1) , (3, 1) , (2, 2) , (1, 2) , (1, 1)]
	10	[(2, 2) , (3, 1) , (2, 1) , (1, 2) , (1, 1)]

ทำการตัดโครโมโซมที่ซ้ำกันออก จากนั้นคำนวณหา ค่าใช้จ่ายรวม และ ระยะทางรวม เพื่อพิจารณาค่าความเหมาะสม

โครโมโซม	เส้นทาง	ค่าใช้จ่าย (บาทต่อวัน)	ระยะทาง (กม.ต่อวัน)	ค่าความเหมาะสม
1	[(2, 2) , (2, 1) , (3, 1) , (1, 2) , (1, 1)]	35,600	1,799.8	3
2	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]	35,600	1,799.8	3
3	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]	35,600	1,799.8	3
4	[(1, 1) , (2, 2) , (3, 1) , (1, 2) , (2, 1)]	38,700	1,924	2
5	[(2, 1) , (1, 1) , (3, 1) , (1, 2) , (2, 2)]	38,700	1,927	2
9	[(2, 1) , (3, 1) , (2, 2) , (1, 2) , (1, 1)]	-	-	1
10	[(2, 2) , (3, 1) , (2, 1) , (1, 2) , (1, 1)]	-	-	1

ทำการจัดลำดับใหม่โดยพิจารณาจากค่าความเหมาะสม จากมากไปน้อย ถ้าค่าความเหมาะสม เท่ากันให้พิจารณาจากค่าใช้จ่ายรวมจากน้อยไปมาก และถ้าค่าใช้จ่ายรวมเท่ากันให้พิจารณาจากค่าระยะทางรวมจากน้อยไปมาก ตามลำดับ

โครโมโซม	เส้นทาง	ค่าใช้จ่าย (บาทต่อวัน)	ระยะทาง (กม.ต่อวัน)	ค่าความเหมาะสม
1	[(2, 2) , (2, 1) , (3, 1) , (1, 2) , (1, 1)]	35,600	1,799.8	3
2	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]	35,600	1,799.8	3
3	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]	35,600	1,799.8	3
4	[(1, 1) , (2, 2) , (3, 1) , (1, 2) , (2, 1)]	38,700	1,924	2
5	[(2, 1) , (1, 1) , (3, 1) , (1, 2) , (2, 2)]	38,700	1,927	2
9	[(2, 1) , (3, 1) , (2, 2) , (1, 2) , (1, 1)]	-	-	1
10	[(2, 2) , (3, 1) , (2, 1) , (1, 2) , (1, 1)]	-	-	1

ทำการคัดเลือกโครโมโซมที่อยู่ใน 5 อันดับแรก (ให้เท่ากับจำนวนโครโมโซมตั้งต้น) เพื่อทำการเก็บค่าที่ดีที่สุดไว้ดำเนินการใน Genetic Algorithm ในรอบถัดไป

โครโมโซมที่ดีที่สุดในรอบที่สองและเป็นโครโมโซมตั้งต้นในรอบถัดไป

โครโมโซม	เส้นทาง
1	[(2, 2) , (2, 1) , (3, 1) , (1, 2) , (1, 1)]
2	[(2, 1) , (2, 2) , (3, 1) , (1, 2) , (1, 1)]
3	[(2, 1) , (2, 2) , (3, 1) , (1, 1) , (1, 2)]
4	[(1, 1) , (2, 2) , (3, 1) , (1, 2) , (2, 1)]
5	[(2, 1) , (1, 1) , (3, 1) , (1, 2) , (2, 2)]

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ข
รหัสคำสั่งที่ใช้ในโปรแกรม IBM ILOG CPLEX OPTIMIZATION STUDIO สำหรับการสร้างแบบจำลองทางคณิตศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

1. รหัสคำสั่งในส่วนของแบบจำลองทางคณิตศาสตร์

```
int numnode = ... ;
int numtruck = ... ;
int numtrip = ... ;
int numedge = ... ;

range node = 0..numnode ;
range truck = 1..numtruck ;
range trip = 1..numtrip ;
range edge = 1..numedge ;

float distance[node][node] =... ;
float time_to_travel[node][node] =... ;
float weight_of_product[node] =... ;
float time_to_load =... ;
float capacity[truck] =... ;
float M = ... ;
float limit_time = ... ;
float distanceedge[edge] = ... ;
float costedge[edge] = ... ;

dvar boolean X[node][node][truck][trip] ;
dvar float+ beg[truck][trip] ;
dvar float+ end[truck][trip] ;
dvar float+ WV[node][truck][trip] ;
dvar float+ CWV[node][node][truck][trip] ;
dvar int+ u[node][truck][trip] ;
dvar float+ SumWV[node] ;
dvar float+ SumTruck[truck][trip] ;
dvar boolean zedge[truck][trip][edge] ;
dvar float+ totaltripdistance[truck][trip] ;

minimize
        sum(c in truck, r in trip, e in edge)costedge[e]*zedge[c][r][e] ;
```

```
subject to {
        forall(i in node)
        ct1 : sum(r in trip,c in truck) WV[i][c][r]>= weight_of_product[i] ;

        forall(i in 1..numnode,j in 1..numnode, c in truck, r in trip)
        ct2 : u[i][c][r] - u[j][c][r]+numnode*X[i][j][c][r] <= numnode-1 ;

        forall(i in node,c in truck,r in trip)
        ct3 : sum(j in node)X[i][j][c][r] - sum(k in node)X[k][i][c][r] == 0 ;

        forall(j in node,c in truck,r in trip)
        ct4 : sum(i in node:i!=j) CWV[i][j][c][r] + WV[j][c][r] <= sum(k in
    node:k!=j)CWV[j][k][c][r] + M*X[0][j][c][r] ;

        forall(i in node,j in node,c in truck, r in trip)
        ct5 : CWV[i][j][c][r] <= capacity[c];

        forall(j in node,c in truck, r in trip)
        ct6 : WV[j][c][r] <= capacity[c]*sum(i in node)X[j][i][c][r] ;

        forall(c in truck,r in trip)
        ct7 : sum(j in node,i in node)time_to_travel[i][j]*X[i][j][c][r] +sum(j in
    node) time_to_load*WV[j][c][r]+beg[c][r] == end[c][r] ;

        forall(c in truck,r in 1..numtrip-1)
        ct8 : beg[c][r+1] >= end[c][r] ;

        forall(c in truck,r in trip)
        ct9 : end[c][r] <= limit_time ;

        forall(c in truck,r in 1..numtrip-1)
        ct10 : sum(j in node)X[0][j][c][r] >= sum(j in node)X[0][j][c][r+1] ;

        forall(i in node)
        ct11 : SumWV[i] == sum(c in truck, r in trip)WV[i][c][r];
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        forall(c in truck, r in trip)
        ct12 : SumTruck[c][r] == sum(i in node)WV[i][c][r];

        forall(c in truck, r in trip)
        ct13 : SumTruck[c][r] <= capacity[c];

        forall(c in truck, r in trip)
        ct14 : sum(i in node,j in node)distance[i][j]*X[i][j][c][r] <= sum(e in
edge)distanceedge[e]*zedge[c][r][e];

        forall(c in truck, r in trip)
        ct15 : sum(e in edge)zedge[c][r][e] <= 1;

        forall(c in truck, r in trip)
        ct16 : sum(i in node,j in node)distance[i][j]*X[i][j][c][r] <=
distanceedge[numedge];


        forall(c in truck, r in trip)
        ct17 : totaltripdistance[c][r] == sum(i in node,j in
node)distance[i][j]*X[i][j][c][r];

        forall(c in truck, r in trip)
        ct18 : X[0][0][c][r] == 0;
    }
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2. รหัสคำสั่งในส่วนของข้อมูลนำเข้า (ตัวอย่าง การไปรับสินค้าทั้งหมด 5 จุด กรณีโรงกำจัดซากไก่ตั้งที่ ตำบลลำพญากลาง จังหวัดลพบุรี)

```
numnode = 3 ;
numtruck = 5 ;
numtrip = 4 ;
numedge = 10;

distance = [[0, 226, 257, 276, 322, 316],
            [226, 0, 67.1, 128, 115, 103],
            [257, 67.1, 0, 120, 151, 136],
```

```
                [276, 128, 120, 0, 236, 224],
                [322, 115, 151, 236, 0, 59.7]
                [316, 103, 136, 224, 59.7, 0]] ;
time_to_travel = [[0, 226, 257, 276, 322, 316],
                  [226, 0, 67.1, 128, 115, 103],
                  [257, 67.1, 0, 120, 151, 136],
                  [276, 128, 120, 0, 236, 224],
                  [322, 115, 151, 236, 0, 59.7]
                  [316, 103, 136, 224, 59.7, 0]] ;
weight_of_product = [0, 5.22, 1.74, 9.86, 2.90, 3.19] ;
time_to_load = 20 ;
capacity = [12, 12, 12, 12, 12] ;
M = 1000;
limit_time = 1440;
distanceedge = [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000];
costedge = [3740, 5270, 6700, 8300, 9800, 11300, 13000, 14400, 16000, 17500];
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก ค
รหัสคำสั่งของ Saving Algorithm และ Genetic Algorithm ที่ใช้ใน JAVA Script

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1. รหัสคำสั่งการจัดรูปแบบของ interface ตัวโปรแกรม

```
<!DOCTYPE html>

<html lang="en">
<head>
   <meta charset="utf-8" />
   <title>CHICKEN RENDERING ROUTING</title>
   <link rel="stylesheet" href="app.css" type="text/css" />
   <script src="jquery-2.0.3.js"></script>
</head>
<body>
   <div id="div_container">

      <h1>CHICKEN RENDERING ROUTING</h1>

               <!-- จำนวน node -->
      <div id="div_input_data">
         <h1>จำนวนโหนด</h1>
         <input type="number" id="input_node_num" min="2" value="22" />
         <br><br>
          <h1>พื้นที่โรงกำจัดซาก</h1>
         <select name="depot" id="input_depot">
                              <option value="depot_lopburi" selected>ลพบุรี
</option>
                              <option value="depot_tubkwang">ทับกวาง</option>
                        </select>
                        <br><br>
         <input type="button" class="btn" id="input_node_btn" value="NEXT"/>
      </div>
               <br>

               <!-- ระยะทางระหว่าง node -->
      <div id="div_input_table" class="hide_at_start clear_at_start">
      </div>
               <br>

               <!-- น้ำหนักของที่แต่ละ node -->
      <div id="div_input_weight" class="hide_at_start clear_at_start">
      </div>
               <br>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                <!-- คุณสมบัติรถที่ใช้ -->
        <div id="div_input_car_detail" class="hide_at_start">
            <h1>Truck Information</h1>
            <select name="car_type" id="input_car_type">
                <option value="" disabled="disabled" selected="selected">เลือกชนิดรถที่
ใช้</option>
                                <option value="car_10">รถสิบล้อ ขนาดบรรทุก 12 ตัน
</option>
                                <option value="car_6">รถหกล้อ ขนาดบรรทุก 7 ตัน
</option>
                                <option value="car_4">รถสี่ล้อ ขนาดบรรทุก 2 ตัน
</option>
                            </select>
            <table>
                <tr>
                    <td>Capacity :</td>
                    <td><input type="number" id="input_car_capacity" min="0"/></td>
                    <td>ton</td>
                </tr>
                <tr>
                    <td>Speed :</td>
                    <td><input type="number" id="input_car_speed" min="0"/></td>
                    <td>km/hr</td>
                </tr>
                <tr>
                    <td>Time to Load :</td>
                    <td><input type="number" id="input_car_load_time" min="0"/></td>
                    <td>min/ton</td>
                </tr>
                <tr>
                    <td>Time Limit :</td>
                    <td><input type="number" id="input_car_limit_time" min="0"/></td>
                    <td>min</td>
                </tr>
            </table>
        </div>
                <br>

                <!-- ต้นทุนค่าขนส่ง -->
        <div id="div_input_car_cost" class="hide_at_start">
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        <h1>Cost Information</h1>
        <input type="button"  class="btn" id="input_cost_btn" value="Add Cost"/>
        <div id="div_cost" class="clear_at_start"></div>
    </div>
                <br>

                <!--เริ่มคำนวณหาค่า Saving -->
    <div id="div_start_btn" class="hide_at_start">
        <input type="button"  class="btn" id="input_all_btn" value="SHOW RESULT" />
    </div>
                <br>

                <!-- ตาราง Saving -->
    <div id="div_output_saving_table" class="hide_at_start clear_at_start">
    </div>
                <br>

                <!-- ผลลัพธ์เส้นทางจาก Saving Heuristic -->
    <div id="div_output_table" class="hide_at_start clear_at_start">
    </div>
                <br>

                <!-- ตั้งค่า parameter สำหรับ Genetic Algorithm -->
    <div id="div_input_genetic" class="hide_at_start">
        <h1>Genetic Algorithm Setup</h1>
        <table>
            <tr>
                <td>จำนวน String คำตอบเริ่มต้น ไม่เกิน :</td>
                <td><input type="number" id="input_num_init_string" min="0" value="10"
/></td>
                <td>Strings</td>
            </tr>
            <tr>
                <td>CrossOver Prob. :</td>
                <td><input type="number" id="input_crossover_prob" min="0" value="0.7"
/></td>
                <td>0-1</td>
            </tr>
            <tr>
                <td>Mutation Prob. :</td>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                <td><input type="number" id="input_mutation_prob" min="0" value="0.3"
/></td>
                <td>0-1</td>
            </tr>
            <tr>
                <td>จำนวนรอบการทำงาน :</td>
                <td><input type="number" id="input_num_loop" min="0" value="10"
/></td>
                <td>ครั้ง</td>
            </tr>
            <tr>
                <td>หยุดเมื่อพบคำตอบที่ดีที่สุดซ้ำกัน :</td>
                <td><input type="number" id="input_num_best_answer" min="0"
value="10" /></td>
                <td>ครั้ง</td>
            </tr>
        </table>
    </div>
    <br>

    <!--เริ่มคำนวณหา Genetic -->
    <div id="div_genetic_start_btn" class="hide_at_start">
        <input type="button"  class="btn" id="input_genetic_btn" value="SHOW RESULT"
/>
    </div>

                <!--เริ่มคำนวณหา Genetic -->
    <div id="div_genetic_init_string" class="hide_at_start">

    </div>

                <!--เริ่มคำนวณหา Genetic -->
    <div id="div_genetic_string" class="hide_at_start">

    </div>

                <!--เริ่มคำนวณหา Genetic -->
    <div id="div_test" class="hide_at_start">

    </div>
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        <!-- Clear All Data -->
        <div id="div_clear" class="hide_at_start">
            <input type="button"  class="btn" id="clear_btn" value="CLEAR"/>
        </div>

    </div>

    <script src="app.js"></script>
    <script src="genetic.js"></script>
</body>
</html>
```

2. รหัสคำสั่งการจัดรูปแบบของการแสดงผลตัวโปรแกรม

```
* {
    margin:0px;
    padding:0px;
 }
title
{
text-align:center;
}
h1
{

            font: 900 30px "Arial", Helvetica, sans-serif;
font-weight: bold;
  color: #0000CD;
  font-size: 32px;
}
body
{
    font-family: 'Segoe UI', sans-serif
}

table
{
margin:0px;padding:0px;
            width:100%;
            box-shadow: 10px 10px 5px #888888;
            border:1px solid #ffffff;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
}
span {
    font-style: italic
}

th {
   background:-o-linear-gradient(bottom, #0057af 5%, #0057af 100%);          background:-
webkit-gradient( linear, left top, left bottom, color-stop(0.05, #0057af), color-stop(1,
#0057af) );
          background:-moz-linear-gradient( center top, #0057af 5%, #0057af 100% );
          filter:progid:DXImageTransform.Microsoft.gradient(startColorstr="#0057af",
endColorstr="#0057af");   background: -o-linear-gradient(top,#0057af,0057af);

          background-color:#0057af;
          border:0px solid #ffffff;
          text-align:center;
          border-width:0px 0px 1px 1px;
          font-size:24px;
          font-family:Arial;
          font-weight:bold;
          color:#ffffff;
}
input
{
width:100%;
}
td
{
 border:1px solid gray;

          vertical-align:middle;

          background-color:#aad4ff;

          border:1px solid #ffffff;
          border-width:0px 1px 1px 0px;
          text-align:left;
          padding:7px;
          font-size:24px;
          font-family:Arial;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            font-weight:normal;
            color:#000000;
}
td, td input {
    font-weight:600;
    text-align:right;
}

th, td, input, .btn {

    height:30px;
}

.btn {
    padding:5px;
}

div {
    margin: 10px 0px;
}

#div_container {
    margin:20px;
}

.td_running_no {
    text-align:center;
}

#div_input_table td, #div_input_table td input , #div_output_saving_table td,
#div_output_saving_table td input {
    width:60px;
}

.td_disable {
    background-color:gray;
    color:white;
}

.align_left {
    text-align:left;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
}

.CSSTableGenerator {
	margin:0px;padding:0px;
	width:100%;
	box-shadow: 10px 10px 5px #888888;
	border:1px solid #ffffff;

	-moz-border-radius-bottomleft:0px;
	-webkit-border-bottom-left-radius:0px;
	border-bottom-left-radius:0px;

	-moz-border-radius-bottomright:0px;
	-webkit-border-bottom-right-radius:0px;
	border-bottom-right-radius:0px;

	-moz-border-radius-topright:0px;
	-webkit-border-top-right-radius:0px;
	border-top-right-radius:0px;

	-moz-border-radius-topleft:0px;
	-webkit-border-top-left-radius:0px;
	border-top-left-radius:0px;
}.CSSTableGenerator table{
	width:100%;
	height:100%;
	margin:0px;padding:0px;
}.CSSTableGenerator tr:last-child td:last-child {
	-moz-border-radius-bottomright:0px;
	-webkit-border-bottom-right-radius:0px;
	border-bottom-right-radius:0px;
}
.CSSTableGenerator table tr:first-child td:first-child {
	-moz-border-radius-topleft:0px;
	-webkit-border-top-left-radius:0px;
	border-top-left-radius:0px;
}
.CSSTableGenerator table tr:first-child td:last-child {
	-moz-border-radius-topright:0px;
	-webkit-border-top-right-radius:0px;
	border-top-right-radius:0px;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
}.CSSTableGenerator tr:last-child td:first-child{
	-moz-border-radius-bottomleft:0px;
	-webkit-border-bottom-left-radius:0px;
	border-bottom-left-radius:0px;
}.CSSTableGenerator tr:hover td{
	background-color:#d3e9ff;


}
.CSSTableGenerator td{
	vertical-align:middle;

	background-color:#aad4ff;

	border:1px solid #ffffff;
	border-width:0px 1px 1px 0px;
	text-align:left;
	padding:7px;
	font-size:10px;
	font-family:Arial;
	font-weight:normal;
	color:#000000;
}.CSSTableGenerator tr:last-child td{
	border-width:0px 1px 0px 0px;
}.CSSTableGenerator tr td:last-child{
	border-width:0px 0px 1px 0px;
}.CSSTableGenerator tr:last-child td:last-child{
	border-width:0px 0px 0px 0px;
}
.CSSTableGenerator tr:first-child td{
		background:-o-linear-gradient(bottom, #0057af 5%, #0057af 100%);
	background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05,
#0057af), color-stop(1, #0057af) );
	background:-moz-linear-gradient( center top, #0057af 5%, #0057af 100% );
	filter:progid:DXImageTransform.Microsoft.gradient(startColorstr="#0057af",
endColorstr="#0057af");  background: -o-linear-gradient(top,#0057af,0057af);

	background-color:#0057af;
	border:0px solid #ffffff;
	text-align:center;
	border-width:0px 0px 1px 1px;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
	font-size:14px;
	font-family:Arial;
	font-weight:bold;
	color:#ffffff;
}
.CSSTableGenerator tr:first-child:hover td{
	background:-o-linear-gradient(bottom, #0057af 5%, #0057af 100%);  background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #0057af), color-stop(1, #0057af) );
	background:-moz-linear-gradient( center top, #0057af 5%, #0057af 100% );
	filter:progid:DXImageTransform.Microsoft.gradient(startColorstr="#0057af", endColorstr="#0057af");  background: -o-linear-gradient(top,#0057af,0057af);

	background-color:#0057af;
}
.CSSTableGenerator tr:first-child td:first-child{
	border-width:0px 0px 1px 0px;
}
.CSSTableGenerator tr:first-child td:last-child{
	border-width:0px 0px 1px 1px;
}


/*input{

background:#2daebf;

font-family:Tahoma, Geneva, sans-serif;

font-size:16px;

padding: 5px 10px 5px 10px;

color: #fff;

font-weight: bold;

-moz-border-radius: 6px;

-webkit-border-radius: 6px;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
-moz-box-shadow: 0 1px 3px rgba(0,0,0,0.5);

-webkit-box-shadow: 0 1px 3px rgba(0,0,0,0.5);

text-shadow: 0 -1px 1px rgba(0,0,0,0.25);

border-bottom: 1px solid rgba(0,0,0,0.25);

cursor: pointer;

border-left:none;

border-top:none;

margin:10px 0 10px 0;

}

input:hover{

background:#49C4D4;

}*/

input {
        font-size:20px;
}
select {
        font-size: 20px;
         height:30px;
         width: 100%;
}
```

3. รหัสคำสั่งการทำงานทั้งหมดตั้งแต่การกำหนดค่าพามิเตอร์ของข้อมูลนำเข้าจนถึงจบกระบวนการ Saving Algorithm

```
var nodeCount; // จำนวน node ทั้งหมด

var capacity;
var speed;
var loadTime;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
var limitedTime;

/// <reference path="jquery.d.ts" />
var Matrix = (function () {
    function Matrix(rows, columns) {
        this.rows = rows;
        this.columns = columns;
        this.datas = new Array(rows);
        for (var i = 0; i < rows; i++) {
            this.datas[i] = new Array(columns);
        }
    }
    return Matrix;
})();


var Route = (function(){
            function Route(carResults, savingObj){
                        this.carResults = carResults;
                        this.savingObj = savingObj;

                        /*
                        var allDistance, allTime, allCost;
                        allDistance = allTime = allCost = 0;

                        for (var i = 0; i < this.carResults.length; i++) {
            var carResult = carResults[i];
            for (var i = 0; i < carResult.travels.length; i++) {
                        var travel = carResult.travels[j];

                        allDistance += travel.distance;
                        allTime += travel.time;
                        allCost += travel.cost;
            } // end for j
            result += this.travels[i].time;
        }   // end for i
        */
            }

            Route.prototype.getCarResults = function () {
        return this.carResults;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    };

    Route.prototype.getFuck = function () {
        return "FUCK";
    };
})();

var initRoute = new Array();

var CarResult = (function () {
            // carResult = carResults[i]
            // travel = carResult.travels[j]
            // path = travel.path[k]
    function CarResult() {
    }
    CarResult.prototype.getTotalTime = function () {
        var result = 0;
        for (var i = 0; i < this.travels.length; i++) {
            result += this.travels[i].time;
        }
        return result;
    };
    CarResult.prototype.getTotalDistance = function () {
        var result = 0;
        for (var i = 0; i < this.travels.length; i++) {
            result += this.travels[i].distance;
        }
        return result;
    };
    CarResult.prototype.getTotalCost = function (savingObj) {
        var result = 0;
                    for (var z = 0; z < this.travels.length; z++) {
            result +=  savingObj.getCost(this.travels[z].distance);
        }
        return result;
    };
    return CarResult;
})();

var Traveling = (function () {
    function Traveling() {
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
    }
    return Traveling;
})();

var SavingData = (function () {
    function SavingData(row, column, value) {
        this.row = row;
        this.column = column;
        this.value = value;
        this.isUsed = false;
    }
    return SavingData;
})();

var Cost = (function () {
    function Cost(from, to, cost) {
        this.rangeFrom = from;
        this.rangeTo = to;
        this.cost = cost;
    }
    return Cost;
})();



var Saving = (function () {
    function Saving(node, depot) {
        this.clearAllData();
        this.node = parseInt(node.toString()) + 1;
        this.distanceMatrix = new Matrix(this.node, this.node);
        this.depot = depot;
    }

    Saving.prototype.clearAllData = function () {
        this.customerNeeds = [];
        this.costs = new Array();
    };

    Saving.prototype.gatherRequiredData = function (carCapacity, carSpeed, limitTime,
timeToLoad) {
        this.carCapacity = carCapacity;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    this.carSpeed = carSpeed;
    this.limitTime = limitTime;
    this.timeToLoad = timeToLoad;
    this.carMilePermin = carSpeed / 60;
};
Saving.prototype.gatherDistanceData = function (value, indexX, indexY) {
    this.distanceMatrix.datas[indexX][indexY] = value;
};
Saving.prototype.gatherCustomerNeedData = function (weight) {
    this.customerNeeds.push(weight);
};
Saving.prototype.gatherCost = function (cost) {
    this.costs.push(cost);
};

Saving.prototype.getCost = function (distance) {
    this.costs.sort(function (object1, object2) {
        return object1.rangeFrom - object2.rangeFrom;
    });
    for (var i = 0; i < this.costs.length; i++) {
        var c = this.costs[i];
        if (distance > c.rangeFrom && distance <= c.rangeTo) {
            return c.cost;
        }
    }
    return this.costs[this.costs.length - 1].cost;
};


Saving.prototype.generateInputDistanceTable = function (elementId) {
    $("#" + elementId).html("");

    var myH1 = document.createElement("h1");
    myH1.innerText = "Distance Table";
    $("#" + elementId).append(myH1);

                //alert( this.depot.toString());

                if(  this.depot.toString() == "depot_lopburi"){
        var defaultDistance =
        [226,     // 1
```

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

```
257, 67.1,
276, 128, 120,
322, 115, 151, 236,
316, 103, 136, 224, 59.7, // 5
271, 73.4, 132, 212, 41.7, 59,
244, 64.6, 37.2, 87.2, 163, 139, 134,
130, 273, 319, 331, 276, 296, 253, 307,
120, 274, 309, 321, 258, 278, 235, 296, 12.2,
75.6, 229, 259, 277, 284, 304, 261, 252, 59.2, 61.1, // 10
188, 47.4, 123, 156, 127, 120, 89.1, 119, 222, 223, 181,

189, 24.9, 77.8, 124, 136, 126, 114, 79, 233, 235, 192, 47.7,
191, 83.7, 82.6, 85.4, 199, 182, 170, 75.2, 258, 260, 218, 102, 69.7,
123, 282, 318, 330, 392, 373, 327, 305, 238, 240, 198, 244, 245, 247,
48.2, 184, 213, 231, 293, 274,  228, 206, 156, 158, 116, 145, 146, 148, 109, //15
250, 40.4, 58.3, 152, 99.2, 79.4, 73.7, 68.3, 269, 295, 253, 82.4, 58, 106, 305, 215,
247, 67.6, 36.6, 86.9, 162, 138, 137, 3, 291, 293, 250, 117, 76.5, 72.6, 302, 203,
74.3,
109, 121, 151, 168, 222, 211, 194, 143, 153, 155, 113, 76.9, 83.3, 107, 178, 79.4,
146, 139,
84.9, 146, 176, 193, 247, 237, 191, 169, 194, 195, 153, 108, 109, 110, 140, 40.9,
171, 164, 46.6,
104, 134, 170, 182, 244, 225, 179, 157, 182, 184, 141, 96.2, 96.8, 88.2, 159, 60,
159, 152, 34, 25.4, //20
64.9, 170, 199, 217, 270, 260, 214, 192, 150, 152, 110, 120, 132, 156, 140, 44.3,
194, 187, 42.6, 34.4, 44,
255, 137, 128, 16.6, 250, 231, 219, 96, 323, 330, 282, 166, 134, 72.8, 320, 212,
165, 95.4, 189, 177, 161, 218];
        }
            else if( this.depot.toString() == "depot_tubkwang"){
                var defaultDistance =
[131, // 1
166, 67.1,
181, 128, 120,
233, 115, 151, 236,
221, 103, 136, 224, 59.7, // 5
209, 73.4, 132, 212, 41.7, 59,
149, 64.6, 37.2, 87.2, 163, 139, 134,
162, 273, 319, 331, 276, 296, 253, 307,
152, 274, 309, 321, 258, 278, 235, 296, 12.2,
107, 229, 259, 277, 284, 304, 261, 252, 59.2, 61.1, // 10
```

```
        82.8, 47.4, 123, 156, 127, 120, 89, 119, 222, 223, 181,
        94.1, 24.9, 77.8, 124, 136, 126, 114, 79, 223, 235, 192, 47.7,
        119, 83.7, 82.6, 85.4, 199, 182, 170, 75, 258, 260, 218, 102, 69.7,
        191, 282, 318, 330, 392, 373, 327, 305, 238, 240, 198, 244, 245, 247,
        92.1, 184, 213, 231, 293, 274, 228, 206, 156, 158, 116, 145, 146, 148, 109, // 15
        155, 40.4, 58.3, 152, 99.2, 79, 74, 68, 269, 295, 253, 82.4, 58, 106, 305, 215
                152, 67.6, 36.6, 86.9, 162, 138, 137, 3, 291, 293, 250, 117, 76.5, 72.6, 302,
203, 74.3,
        14.5, 121, 151, 168, 222, 211, 194, 143, 153, 155, 113, 76.9, 83.3, 107, 178, 79.4,
146, 139,
        54.8, 146, 176, 193, 247, 237, 191, 169, 194, 195, 153, 108, 109, 110, 140, 40.9,
171, 164, 46.6,
        43, 134, 170, 182, 244, 225, 179, 157, 182, 184, 141, 9632, 96.8, 88.2, 159, 60,
159, 152, 34, 25.4, // 20
        30.9, 170, 199, 217, 270, 260, 214, 192, 150, 152, 110, 120, 132, 156, 140, 44.3,
194, 187, 42.6, 34.4, 44,
        190, 137, 128, 16.6, 250, 231, 219, 96, 323, 330, 282, 166, 134, 72.8, 320, 212,
165, 95.4, 189, 177, 161, 218];
                }

    var defaultCounter = 0;

    var loop = parseInt(this.node.toString());
    var table = document.createElement("table");
    table.id = "input_table_dist";
    var trh = document.createElement("tr");
    for (var n = 0; n < loop + 1; n++) {
        var th = document.createElement("th");
        th.innerText = n > 0 ? (n == 1 ? "Depot" : (n - 1).toString()) : "";
        trh.appendChild(th);
        table.appendChild(trh);
    }

    for (var i = 0; i < loop; i++) {
        var tr = document.createElement("tr");
        for (var j = 0; j <= i + 1; j++) {
            var td = document.createElement("td");
            if (j == 0) {
                td.innerText = i == 0 ? "Depot" : (i).toString();
                td.setAttribute("class", "td_running_no");
            } else {
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                if (i == j - 1) {
                    td.setAttribute("class", "td_disable");
                    td.innerHTML = "0";
                } else {
                    var input = document.createElement("input");
                    input.type = "number";
                    input.id = "input_dist_" + (j - 1).toString() + "_" + (i).toString();
                    input.setAttribute("xVal", i.toString());
                    input.setAttribute("yVal", (j - 1).toString());
                    if (defaultCounter < defaultDistance.length) {
                        input.value = defaultDistance[defaultCounter].toString();
                        defaultCounter++;
                    }
                    input.min = "0";
                    td.appendChild(input);
                }
            }
            tr.appendChild(td);
        }
        for (var k = i; k < loop - 1; k++) {
            var td = document.createElement("td");
            td.setAttribute("class", "td_disable");
            tr.appendChild(td);
        }
        table.appendChild(tr);
    }
    $("#" + elementId).append(table);
};
Saving.prototype.generateInputWeight = function (elementId) {
    $("#" + elementId).html("");

    var myH1 = document.createElement("h1");
    myH1.innerText = "Weight Information";
    $("#" + elementId).append(myH1);

    var table = document.createElement("table");
    table.id = "input_table_weight";
    var trWH = document.createElement("tr");
    var headerWs = ["NODE", "WEIGHT(T)"];
    for (var i = 0; i < headerWs.length; i++) {
        var th = document.createElement("th");
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            th.innerText = headerWs[i];
            trWH.appendChild(th);
        }
        table.appendChild(trWH);


        var weights = [5.22, 1.74, 9.86, 2.90, 3.19, 3.77, 2.03, 7.54, 3.19, 3.77, 2.61, 2.61, 2.61,
7.25, 8.99, 3.77, 3.77, 6.96, 7.83, 6.38, 2.61, 1.45];
        for (var i = 0; i < this.node - 1; i++) {
            var tr = document.createElement("tr");

            var tdNum = document.createElement("td");
            tdNum.innerText = (i + 1).toString();
            tdNum.setAttribute("class", "td_running_no");
            tr.appendChild(tdNum);

            var tdWeight = document.createElement("td");
            var input = document.createElement("input");
            input.type = "number";
            input.id = "input_weight_" + i.toString();
            input.min = "0";
            if (i < weights.length)
                input.value = weights[i].toString();
            tdWeight.appendChild(input);
            tr.appendChild(tdWeight);

            table.appendChild(tr);
        }
        $("#" + elementId).append(table);
    };

    // สร้างช่องกรอกข้อมูล ต้นทุนค่าขนส่ง
    Saving.prototype.generateInputCost = function (elementId) {
        $("#" + elementId).html("");

        var table = document.createElement("table");
        table.id = "input_table_cost";
        var trWH = document.createElement("tr");
        var headerWs = ["Start", "To", "Cost"];
        for (var i = 0; i < headerWs.length; i++) {
            var th = document.createElement("th");
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        th.innerText = headerWs[i];
        trWH.appendChild(th);
    }
    table.appendChild(trWH);

    var tr = document.createElement("tr");
    tr.id = "input_cost_row_" + table.childElementCount.toString();
    var tdFrom = document.createElement("td");
     {
        var input = document.createElement("input");
        input.type = "number";
        input.min = "0";
        input.id = "input_cost_from_" + table.childElementCount.toString();
        tdFrom.appendChild(input);
    }
    var tdTo = document.createElement("td");
     {
        var input = document.createElement("input");
        input.type = "number";
        input.min = "0";
        input.id = "input_cost_to_" + table.childElementCount.toString();
        tdTo.appendChild(input);
    }
    var tdCost = document.createElement("td");
     {
        var input = document.createElement("input");
        input.type = "number";
        input.min = "0";
        input.id = "input_cost_value_" + table.childElementCount.toString();
        tdCost.appendChild(input);
    }
    tr.appendChild(tdFrom);
    tr.appendChild(tdTo);
    tr.appendChild(tdCost);
    table.appendChild(tr);

    $("#" + elementId).append(table);
};
Saving.prototype.generateInput = function (distanceTableId, weightId, costId) {
    this.generateInputDistanceTable(distanceTableId);
    this.generateInputWeight(weightId);
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    this.generateInputCost(costId);
};

    // Saving Method generateSavingMatrix - คำนวณหา Saving Table
Saving.prototype.generateSavingMatrix = function () {
    var node = parseInt(this.node.toString()) - 1;
    this.savingMatrix = new Matrix(node, node);
    for (var i = 0; i < node; i++) {
        for (var j = 0; j <= i; j++) {
            this.savingMatrix.datas[i][j] = i == j ? 0 : this.distanceMatrix.datas[i + 1][0] +
this.distanceMatrix.datas[j + 1][0] - this.distanceMatrix.datas[i + 1][j + 1];
        }
    }
};


// Saving Method generateCarResult - คำนวณหา String Result
Saving.prototype.generateCarResult = function () {
    var loop = this.savingMatrix.rows;
    var resultCount = ((loop * loop) - loop) / 2;
    //orderResult => Saving Matrix ที่เพิ่มข้อมูลเข้าไป
    var orderResult = new Array();

    for (var i = 0; i < loop; i++) {
        for (var j = 0; j <= i; j++) {
            if (i != j)
                orderResult.push(new SavingData(i, j, this.savingMatrix.datas[i][j]));
        }
    }

    // sort by saving value [Sort numbers (numerically and descending):]
    orderResult.sort(function (object1, object2) {
        return object2.value - object1.value;
    });

                /*
                // Show orderResult Data inside
                var tmp = "";
                for (var i = 0; i < orderResult.length; i++) {

              tmp += orderResult[i].value + ", ";
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
}
            alert(tmp);
            */

// keep data and create Path
var isUsed = [];
for (var i = 0; i < loop; i++) {
    isUsed[i] = false;
}
var isFinishAllNode = false;
var tmpTravels = new Array();
var counter = 0;
do {
    var tmpTraveling = new Traveling();
    tmpTraveling.time = 0;
    tmpTraveling.weight = 0;
    tmpTraveling.path = [];
    tmpTraveling.weightPath = [];
    for (var i = 0; i < resultCount; i++) {
        var obj = orderResult[i];

        // usage Time = distance * 60 / speed
        var weight = this.customerNeeds[obj.row] + this.customerNeeds[obj.column];

        /*
        // Test weight
        alert("weight = " + weight + "\n" +
                "cneed obj.row = " + this.customerNeeds[obj.row] + "\n" +
                "cneed obj.column = " + this.customerNeeds[obj.column] + "\n"
        );
        */
        var distance = this.distanceMatrix.datas[obj.row + 1][0] +
this.distanceMatrix.datas[obj.row + 1][obj.column + 1] +
this.distanceMatrix.datas[obj.column + 1][0];
        var usageTime = (distance * 60 / this.carSpeed) + (this.timeToLoad *
(this.customerNeeds[obj.row] + this.customerNeeds[obj.column]));
        if ((!isUsed[obj.row] && !isUsed[obj.column]) && (tmpTraveling.time +
usageTime) <= (this.limitTime) && (tmpTraveling.weight + weight) <= this.carCapacity) {
            tmpTraveling.path.push(obj.row + 1);
            tmpTraveling.path.push(obj.column + 1);
            tmpTraveling.time += usageTime;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            tmpTraveling.weight += weight;
            tmpTraveling.weightPath.push(this.customerNeeds[obj.row]);
            tmpTraveling.weightPath.push(this.customerNeeds[obj.column]);
            tmpTraveling.distance = distance;
            isUsed[obj.row] = true;
            isUsed[obj.column] = true;
        }
    }
    isFinishAllNode = tmpTraveling.path.length == 0;
    if (!isFinishAllNode) {
        tmpTravels[counter] = tmpTraveling;
    }
    counter++;
} while(!isFinishAllNode);




for (var i = 0; i < loop; i++) {
    if (!isUsed[i]) {
        var weight = this.customerNeeds[i];
        var distance = this.distanceMatrix.datas[i + 1][0] * 2;
        var usageTime = distance / this.carMilePermin;
        while (weight > this.carCapacity) {
            var tmpTraveling = new Traveling();
            tmpTraveling.path = [];
            tmpTraveling.weightPath = [];
            tmpTraveling.path.push(i + 1);
            tmpTraveling.weight = this.carCapacity;
            tmpTraveling.weightPath.push(this.carCapacity);
            tmpTraveling.time = usageTime + (this.carCapacity * this.timeToLoad);
            tmpTraveling.distance = distance;

            weight = weight - this.carCapacity;

            tmpTravels.push(tmpTraveling);
        }

        if (weight > 0) {
            var tmpTraveling = new Traveling();
            tmpTraveling.path = [];
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                tmpTraveling.weightPath = [];
                tmpTraveling.path.push(i + 1);
                tmpTraveling.time = usageTime + (weight * this.timeToLoad);
                tmpTraveling.weight = weight;
                tmpTraveling.weightPath.push(weight);
                tmpTraveling.distance = distance;

                tmpTravels.push(tmpTraveling);
            }
        }
    }

        // ####### Sort Travel (Path & weightPath in Sequence)

        for(var i=0;i<tmpTravels.length;i++){
            for(var j=0;j<tmpTravels[i].path.length;j++){
                var maxWeight = 0;
                var index = j;
                for(var k=j;k<tmpTravels[i].path.length;k++){
                    if(tmpTravels[i].weightPath[k] > maxWeight){
                        maxWeight =
tmpTravels[i].weightPath[k];
                        index = k;
                    }
                }
                var tmp;
                tmp = tmpTravels[i].path[index];
                tmpTravels[i].path[index] = tmpTravels[i].path[j];
                tmpTravels[i].path[j] = tmp;

                tmp = tmpTravels[i].weightPath[index];
                tmpTravels[i].weightPath[index] =
tmpTravels[i].weightPath[j];
                tmpTravels[i].weightPath[j] = tmp;
            }
        }

        /*
         // The return value of descending (for this example) is important.
    // If the return value is less than zero, the index of a is before b, and
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
// if it is greater than zero it's vice-versa. If the return value is zero, the elements' index is equal.

    function descending( a, b ) {
                        return b - a;
                    }

                    b < a     => b-a => < 0 => return b
                    a < b     => b-a => > 0 => return a
*/

// sort by Capcity (weight)
tmpTravels.sort(function (travel1, travel2) {
    return travel1.weight - travel2.weight;
});

for (var i = 0; i < tmpTravels.length; i++) {
    var index = -1;
    var minDistance = 99999;
    for (var j = 0; j < tmpTravels.length; j++) {
        if (!tmpTravels[i] || !tmpTravels[j])
            continue;

        if (i == j)
            continue;

        if (tmpTravels[i].weight + tmpTravels[j].weight > this.carCapacity)
            continue;

        // calculate for new distance
        var iIndex = tmpTravels[i].path[0];
        var jCount = tmpTravels[j].path.length;
        var jIndex = tmpTravels[j].path[jCount - 1];
        var newDistance = tmpTravels[i].distance + (jIndex > iIndex ? this.distanceMatrix.datas[jIndex][iIndex] : this.distanceMatrix.datas[iIndex][jIndex]) - this.distanceMatrix.datas[jIndex][0] - this.distanceMatrix.datas[iIndex][0];

        if ((tmpTravels[j].distance + newDistance) / this.carMilePermin > this.limitTime)
            continue;

        if (newDistance < minDistance) {
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                index = j;
                minDistance = newDistance;
            }
        }
        if (tmpTravels[i] && index > -1) {
            var sumDistance = tmpTravels[index].distance + minDistance;
            tmpTravels[index].distance = sumDistance;
            tmpTravels[index].weight += tmpTravels[i].weight;
            for (var k = 0; k < tmpTravels[i].path.length; k++) {
                tmpTravels[index].path.push(tmpTravels[i].path[k]);
                tmpTravels[index].weightPath.push(tmpTravels[i].weightPath[k]);
            }
            tmpTravels[index].time = tmpTravels[index].distance / this.carMilePermin;
            for (var k = 0; k < tmpTravels[index].path.length; k++) {
                var indexCust = tmpTravels[index].path[k];
                tmpTravels[index].time += (this.customerNeeds[indexCust - 1] %
this.carCapacity) * this.timeToLoad;
            }
            delete tmpTravels[i];
        }
    }

    // sort by capacity
    tmpTravels.sort(function (travel1, travel2) {
        return travel1.weight - travel2.weight;
    });


    // group to car
    this.carResults = new Array();
    var i = 0;
    while (i < tmpTravels.length) {
        var totalTime = 0;
        var newCarResult = new CarResult();
        newCarResult.travels = new Array();
        for (; i < tmpTravels.length; i++) {
            if (!tmpTravels[i])
                continue;
            if (totalTime + tmpTravels[i].time > this.limitTime)
                break;
            newCarResult.travels.push(tmpTravels[i]);
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            totalTime += tmpTravels[i].time;
        }
        this.carResults.push(newCarResult);
    }


};



Saving.prototype.calculateRoutes = function (x, y) {
};
Saving.prototype.printSavingMatrix = function (elementId) {
    $("#" + elementId).html("");

    var header = document.createElement("h1");
    header.innerText = "Saving Table";
    $("#" + elementId).append(header);

    var table = document.createElement("table");
    table.id = "table_saving_matrix";
    var trHeader = document.createElement("tr");

    var th0 = document.createElement("th");
    th0.innerText = "Dist (km)";
    trHeader.appendChild(th0);

    for (var n = 0; n < this.savingMatrix.columns; n++) {
        var th = document.createElement("th");
        th.innerText = (n + 1).toString();
        trHeader.appendChild(th);
    }
    table.appendChild(trHeader);

    for (var i = 0; i < this.savingMatrix.rows; i++) {
        var tr = document.createElement("tr");
        var td0 = document.createElement("td");
        td0.innerText = (i + 1).toString();
        td0.setAttribute("class", "td_running_no");
        tr.appendChild(td0);
        for (var j = 0; j < this.savingMatrix.columns; j++) {
```

```
                var td = document.createElement("td");
                if (!this.savingMatrix.datas[i][j]) {
                    td.setAttribute("class", "td_disable");
                }
                if (this.savingMatrix.datas[i][j] > -1)
                    td.innerText = this.savingMatrix.datas[i][j].toFixed(1).toString();
                tr.appendChild(td);
            }
            table.appendChild(tr);
        }
        $("#" + elementId).append(table);
    };


    // ##########  Print Result
    Saving.prototype.printCarResult = function (elementId) {

        $("#" + elementId).html("");


        var myH1 = document.createElement("h1");
        myH1.innerText = "Result";
        $("#" + elementId).append(myH1);


        var table = document.createElement("table");
        table.id = "table_result_matrix";

        var trh = document.createElement("tr");
        var header = ["Car No.", "Seq", "Traveling", "Weight", "Distance", "Time", "Cost", "Total
Distance", "Total Time", "Total Cost"];
        for (var i = 0; i < header.length; i++) {
            var th = document.createElement("th");
            th.innerText = header[i];
            trh.appendChild(th);
        }
        table.appendChild(trh);


                    // carResults
                    var allTime = 0;
```

```
                var allDistance = 0;
                var allCost = 0;
                var seq = 0;
                //////////// ########## initRoute
                initRoute['car'] = new Array();


for (var i = 0; i < this.carResults.length; i++) {

    // initRoute
    initRoute['car'][i] = this.carResults[i];

    var carResult = this.carResults[i];
    var travellength = carResult.travels.length;

    for (var j = 0; j < travellength; j++) {
        var tr = document.createElement("tr");
        //Sequence Count
                                seq++;
                                // Column [Car No.]
        if (j == 0) {
            var std0 = document.createElement("td");
            std0.rowSpan = travellength;
            std0.innerText = (i + 1).toString();
            std0.setAttribute("class", "td_running_no");
            tr.appendChild(std0);
        }

        var travel = carResult.travels[j];

        // the cycle of car ...
        // Column [Seq]
        var td0 = document.createElement("td");
        //td0.innerText = (j + 1).toString();
        td0.innerText = (seq).toString();
        tr.appendChild(td0);

                                // Column [Travelling]
        var td1 = document.createElement("td");
        td1.innerText = "Depot --> ";
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
td1.setAttribute("class", "align_left");
for (var k = 0; k < travel.path.length; k++) {
    var path = travel.path[k];
    td1.innerText += (k > 0 ? " --> " : "") + path.toString();
}
td1.innerText += " --> Depot";

td1.innerText += "\n";
td1.innerText += "0 --> ";
for (var k = 0; k < travel.weightPath.length; k++) {
    var weightPath = travel.weightPath[k];
    td1.innerText += (k > 0 ? " --> " : "") + weightPath.toString();
}
td1.innerText += " --> 0";

tr.appendChild(td1);

                        // Column [Weight]
var td2 = document.createElement("td");
td2.innerText = travel.weight.toFixed(2);
tr.appendChild(td2);

                        // Column [Distance]
var td3 = document.createElement("td");
td3.innerText = travel.distance.toFixed(2);
tr.appendChild(td3);

                        // Column [Time]
var td4 = document.createElement("td");
td4.innerText = travel.time.toFixed(2);
tr.appendChild(td4);

                        // Column [Cost]
var td5 = document.createElement("td");
td5.innerText = this.getCost(travel.distance).toFixed(2);
tr.appendChild(td5);



                        // Total Distance Time & Cost
if (j == 0) {
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            // Total for each car ...
            var totalTime = carResult.getTotalTime();
            var totalDistance = carResult.getTotalDistance();
            var totalCost = carResult.getTotalCost(this);

            allTime += totalTime;
            allDistance += totalDistance;
            allCost += totalCost;

            /*
            for (var z = 0; z < travellength; z++) {
                totalCost = totalCost + this.getCost(carResult.travels[z].distance);
            }
                                        */
                                        // Column [Total Distance]
            var std1 = document.createElement("td");
            std1.rowSpan = travellength;
            std1.innerText = totalDistance.toFixed(2);
            tr.appendChild(std1);

                                        // Column [Total Time]
            var std2 = document.createElement("td");
            std2.rowSpan = travellength;
            std2.innerText = totalTime.toFixed(2);
            tr.appendChild(std2);

                                        // Column [Total Cost]
            var std3 = document.createElement("td");
            std3.rowSpan = travellength;
            std3.innerText = totalCost.toFixed(2);
            tr.appendChild(std3);
        }
        table.appendChild(tr);
    }
}



            // Summary Result
            var tr = document.createElement("tr");
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                // Column [Car No.]
var std0 = document.createElement("td");
std0.innerText = "#";
std0.setAttribute("class", "td_running_no");
tr.appendChild(std0);

        // the cycle of car ...
        // Column [Seq]
        var td0 = document.createElement("td");
        td0.innerText = "";
        tr.appendChild(td0);

                // Column [Travelling]
        var td1 = document.createElement("td");
        td1.innerText += "";
        tr.appendChild(td1);

                // Column [Weight]
        var td2 = document.createElement("td");
        td2.innerText = "";
        tr.appendChild(td2);

                // Column [Distance]
var td3 = document.createElement("td");
td3.innerText = "";
tr.appendChild(td3);

                // Column [Time]
var td4 = document.createElement("td");
td4.innerText = "";
tr.appendChild(td4);

                // Column [Cost]
var td5 = document.createElement("td");
td5.innerText = "";
tr.appendChild(td5);

                // Column [Total Distance]
var std1 = document.createElement("td");
std1.rowSpan = travellength;
std1.innerText = allDistance.toFixed(2);
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    tr.appendChild(std1);

                // Column [Total Time]
    var std2 = document.createElement("td");
    std2.rowSpan = travellength;
    std2.innerText = allTime.toFixed(2);
    tr.appendChild(std2);

                // Column [Total Cost]
    var std3 = document.createElement("td");
    std3.rowSpan = travellength;
    std3.innerText = allCost.toFixed(2);
    tr.appendChild(std3);

    table.appendChild(tr);




    $("#" + elementId).append(table);
};

        // Saving Method -
Saving.prototype.printResult = function (savingDivId, resultDivId) {
    this.generateSavingMatrix();
    this.printSavingMatrix(savingDivId);
    this.generateCarResult();
    this.printCarResult(resultDivId);
    this.printRoute();
};

Saving.prototype.printRoute = function(){
        /*
        for(var i=0; i<initRoute['car'].length;i++){
                $("div_genetic").innerHTML += " " +  i + ", ";
        }
        */
        /*
        var tmp ="";
        for(var i=0; i<initRoute['car'][3].travels[0].path.length; i++){
                tmp += initRoute['car'][3].travels[0].path[i] + ", ";
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            }

            //alert(initRoute['car'][1].travels[0].path[i]);
            alert(tmp);
            */
            //$("div_genetic").show();
    };

    return Saving;

})();



// SAVING Object
var myMatrix;

var MIN_NODE_NUM = 2;

$(this).ready(function () {
    initUI();
});

function initUI() {
    $("#div_input_data").show();
    $(".hide_at_start").each(function () {
        if ($(this).hasClass("clear_at_start")) {
            $(this).html("");
        }
        $(this).hide();
    });

    // Default Data:
    $("#input_node_num").val(22);

    $("#input_car_capacity").val(16);
    $("#input_car_speed").val(60);
    $("#input_car_load_time").val(1);
    $("#input_car_limit_time").val(1440);
}
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
function isGatherAllDataComplete() {
    myMatrix.clearAllData();
    var result = true;

    // Distance Data
    $("#input_table_dist input[id^='input_dist_']").each(function () {
        var indexX = parseInt($(this).attr("xVal"));
        var indexY = parseInt($(this).attr("yVal"));
        var value = parseFloat($(this).val());
        if (!value) {
            alert('Please complete distance data.');
            result = false;
            return false;
        } else
            myMatrix.gatherDistanceData(value, indexX, indexY);
    });

/*
    // Car Data
    capacity = parseFloat($("#input_car_capacity").val());
    speed = parseFloat($("#input_car_speed").val());
    loadTime = parseFloat($("#input_car_load_time").val());
    limitedTime = parseFloat($("#input_car_limit_time").val());
 */
    if (!capacity || !speed || !loadTime || !limitedTime) {
        alert('Please input car data.');
        result = false;
    } else
        myMatrix.gatherRequiredData(capacity, speed, limitedTime, loadTime);

    // Weight Data
    $("input[id^='input_weight_']").each(function () {
        var weight = parseFloat($(this).val());
        if (!weight) {
            alert('Please complete requied weights.');
            result = false;
            return false;
        } else
            myMatrix.gatherCustomerNeedData(weight);
    });
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    // Cost Data
    var costCount = 0;
    $("#input_table_cost tr[id^='input_cost_row_']").each(function () {
        var start = parseFloat($(this).find("input[id^='input_cost_from_']").val());
        var to = parseFloat($(this).find("input[id^='input_cost_to_']").val());
        var value = parseFloat($(this).find("input[id^='input_cost_value_']").val());
        if (value > 0) {
            var cost = new Cost(start, to, value);
            myMatrix.gatherCost(cost);
            costCount++;
        }
    });
    if (costCount == 0) {
        alert('Please complete cost data.');
        result = false;
    }

    return result;
}

/*
##########################################################
###
###                          FORM ACTION
###
##########################################################
*/

//
$("#input_node_btn").click(function () {

    nodeCount = $("#input_node_num").val();

    var depot;
    depot = $("#input_depot").val();


    if (nodeCount && nodeCount >= MIN_NODE_NUM) {

        myMatrix = new Saving(nodeCount, depot);
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        // Clear Form in HTML Display
        var ids = ["div_input_table", "div_input_car_detail", "div_input_weight",
"div_input_car_cost", "div_start_btn"];
        for (var i = 0; i < ids.length; i++) {
            if ($("#" + ids[i]).hasClass("clear_at_start")) {
                $("#" + ids[i]).html("");
            }
            $("#" + ids[i]).show();
        }
        $("#div_input_data").hide();


        myMatrix.generateInput("div_input_table", "div_input_weight", "div_cost");
        $("#div_clear").show();

    } else {
        alert("This application does not accept node with less than " +
MIN_NODE_NUM.toString() + ".");
    }
});



$("#clear_btn").click(function () {
    initUI();
});


$("#input_car_type").change(function(){
            var carType;
    carType = $("#input_car_type").val();


    var costList = new Array("from", "to", "cost");
    costList["from"] = new Array();
    costList["to"] = new Array();
    costList["cost"] = new Array();

            if(carType.toString() == "car_10") {
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
$("#input_car_capacity").val(12);
$("#input_car_speed").val(60);
$("#input_car_load_time").val(1);
$("#input_car_limit_time").val(1440);

// 0-100
costList["from"][0] = 0;
costList["to"][0] = 100;
costList["cost"][0] = 3740;

// 100.1-200
costList["from"][1] = 100.1;
costList["to"][1] = 200;
costList["cost"][1] = 5270;

// 200.1-300
costList["from"][2] = 200.1;
costList["to"][2] = 300;
costList["cost"][2] = 6700;

// 300.1-400
costList["from"][3] = 300.1;
costList["to"][3] = 400;
costList["cost"][3] = 8300;

// 400.1-500
costList["from"][4] = 400.1;
costList["to"][4] = 500;
costList["cost"][4] = 9800;

// 500.1-600
costList["from"][5] = 500.1;
costList["to"][5] = 600;
costList["cost"][5] = 11300;

// 600.1-700
costList["from"][6] = 600.1;
costList["to"][6] = 700;
costList["cost"][6] = 13000;

// 700.1-800
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            costList["from"][7] = 700.1;
            costList["to"][7] = 800;
            costList["cost"][7] = 14400;

            // 800.1-900
            costList["from"][8] = 800.1;
            costList["to"][8] = 900;
            costList["cost"][8] = 16000;

            // 900.1-1000
            costList["from"][9] = 900.1;
            costList["to"][9] = 1000;
            costList["cost"][9] = 17500;

}
else if(carType.toString() == "car_6") {
            $("#input_car_capacity").val(7);
            $("#input_car_speed").val(60);
            $("#input_car_load_time").val(1);
            $("#input_car_limit_time").val(1440);

            // 0-100
            costList["from"][0] = 0;
            costList["to"][0] = 100;
            costList["cost"][0] = 2780;

            // 100.1-200
            costList["from"][1] = 100.1;
            costList["to"][1] = 200;
            costList["cost"][1] = 3850;

            // 200.1-300
            costList["from"][2] = 200.1;
            costList["to"][2] = 300;
            costList["cost"][2] = 4920;

            // 300.1-400
            costList["from"][3] = 300.1;
            costList["to"][3] = 400;
            costList["cost"][3] = 5990;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            // 400.1-500
            costList["from"][4] = 400.1;
            costList["to"][4] = 500;
            costList["cost"][4] = 7060;

            // 500.1-600
            costList["from"][5] = 500.1;
            costList["to"][5] = 600;
            costList["cost"][5] = 8130;

            // 600.1-700
            costList["from"][6] = 600.1;
            costList["to"][6] = 700;
            costList["cost"][6] = 9200;

            // 700.1-800
            costList["from"][7] = 700.1;
            costList["to"][7] = 800;
            costList["cost"][7] = 10270;

            // 800.1-900
            costList["from"][8] = 800.1;
            costList["to"][8] = 900;
            costList["cost"][8] = 11340;

            // 900.1-1000
            costList["from"][9] = 900.1;
            costList["to"][9] = 1000;
            costList["cost"][9] = 12420;
    }
    else if(carType.toString() == "car_4") {
            $("#input_car_capacity").val(2);
            $("#input_car_speed").val(60);
            $("#input_car_load_time").val(1);
            $("#input_car_limit_time").val(1440);

            // 0-100
            costList["from"][0] = 0;
            costList["to"][0] = 100;
            costList["cost"][0] = 1700;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
// 100.1-200
costList["from"][1] = 100.1;
costList["to"][1] = 200;
costList["cost"][1] = 2250;

// 200.1-300
costList["from"][2] = 200.1;
costList["to"][2] = 300;
costList["cost"][2] = 2800;

// 300.1-400
costList["from"][3] = 300.1;
costList["to"][3] = 400;
costList["cost"][3] = 3350;

// 400.1-500
costList["from"][4] = 400.1;
costList["to"][4] = 500;
costList["cost"][4] = 3900;

// 500.1-600
costList["from"][5] = 500.1;
costList["to"][5] = 600;
costList["cost"][5] = 4420;

// 600.1-700
costList["from"][6] = 600.1;
costList["to"][6] = 700;
costList["cost"][6] = 4950;

// 700.1-800
costList["from"][7] = 700.1;
costList["to"][7] = 800;
costList["cost"][7] = 5500;

// 800.1-900
costList["from"][8] = 800.1;
costList["to"][8] = 900;
costList["cost"][8] = 6100;

// 900.1-1000
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                costList["from"][9] = 900.1;
                costList["to"][9] = 1000;
                costList["cost"][9] = 6700;
        }


        //var tmp = "";


        var tbl = document.getElementById("input_table_cost");
        tbl.innerHTML = "";


    var trWH = document.createElement("tr");
    var headerWs = ["Start", "To", "Cost"];
    for (var i = 0; i < headerWs.length; i++) {
        var th = document.createElement("th");
        th.innerText = headerWs[i];
        trWH.appendChild(th);
    }
    tbl.appendChild(trWH);


        for(var i=0; i<costList["from"].length; i++){
                //tmp += "From: " + costList["from"][i] + ", To: " + costList["to"][i] + ",
Cost: " + costList["cost"][i] + "\n";

                var tr = document.createElement("tr");
            tr.id = "input_cost_row_" + tbl.childElementCount.toString();
            var tdFrom = document.createElement("td");
             {
                var input = document.createElement("input");
                input.type = "number";
                input.min = "0";
                input.id = "input_cost_from_" + tbl.childElementCount.toString();
                input.value = costList["from"][i];
                tdFrom.appendChild(input);
            }
            var tdTo = document.createElement("td");
             {
                var input = document.createElement("input");
```

```
                input.type = "number";
                input.min = "0";
                input.id = "input_cost_to_" + tbl.childElementCount.toString();
                input.value = costList["to"][i];
                tdTo.appendChild(input);
            }
            var tdCost = document.createElement("td");
             {
                var input = document.createElement("input");
                input.type = "number";
                input.min = "0";
                input.id = "input_cost_value_" + tbl.childElementCount.toString();
                input.value = costList["cost"][i];
                tdCost.appendChild(input);
            }
            tr.appendChild(tdFrom);
            tr.appendChild(tdTo);
            tr.appendChild(tdCost);
            tbl.appendChild(tr);

        }
        //alert(tmp);

});



$("#input_cost_btn").click(function () {
    var tbl = document.getElementById("input_table_cost");
    var tr = document.createElement("tr");
    tr.id = "input_cost_row_" + tbl.childElementCount.toString();
    var tdFrom = document.createElement("td");
     {
        var input = document.createElement("input");
        input.type = "number";
        input.min = "0";
        input.id = "input_cost_from_" + tbl.childElementCount.toString();
        tdFrom.appendChild(input);
    }
    var tdTo = document.createElement("td");
     {
```

```
        var input = document.createElement("input");
        input.type = "number";
        input.min = "0";
        input.id = "input_cost_to_" + tbl.childElementCount.toString();
        tdTo.appendChild(input);
    }
    var tdCost = document.createElement("td");
     {
        var input = document.createElement("input");
        input.type = "number";
        input.min = "0";
        input.id = "input_cost_value_" + tbl.childElementCount.toString();
        tdCost.appendChild(input);
    }
    tr.appendChild(tdFrom);
    tr.appendChild(tdTo);
    tr.appendChild(tdCost);
    tbl.appendChild(tr);
});


$("#input_all_btn").click(function () {

            // Car Data
    capacity = parseFloat($("#input_car_capacity").val());
    speed = parseFloat($("#input_car_speed").val());
    loadTime = parseFloat($("#input_car_load_time").val());
    limitedTime = parseFloat($("#input_car_limit_time").val());

    if (isGatherAllDataComplete()) {
        myMatrix.printResult("div_output_saving_table", "div_output_table");
        $("#div_output_saving_table").show();
        $("#div_output_table").show();
        $("#div_input_genetic").show();
        $("#div_genetic_start_btn").show();
    }
});

/*
#####################                          Genetic
```

```
*/

//@ sourceMappingURL=app.js.map
function log(msg) {
    setTimeout(function() {
        throw new Error(msg);
    }, 0);
};
```

4. รหัสคำสั่งการทำงานทั้งหมดตั้งแต่การกำหนดค่าพามิเตอร์ของ Genetic Algorithm จนถึงจบกระบวนการ

```
/*
############################################################
##
##      Variable
##
############################################################
*/

var geneRoutes = new Array();
var parentGene = new Array();
var matingPool = new Array();
var offspringGene = new Array();
var collectGene = new Array();

var stringLimit;
var crossProb;
var mutationProb;
var geneticLoop;

/*
############################################################
##
##      Class & model
##
############################################################
*/


// Class GeneNode หน่วยย่อยใน GeneString
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
var GeneNode = (function(){
        function GeneNode(car, seq, pos, node, weightPath){
                this.car = car;
                this.node = node;
                this.seq = seq;
                this.pos = pos;
                this.weightPath = weightPath;
        }
});


// Class GeneString
var GeneString = (function () {
    function GeneString(geneNodes) {
        this.nodes = geneNodes; //
    }
})(); // End Class GeneString


//
var PathNode = (function(){
        function PathNode(node, weightPath){
                this.node = node;
                this.weightPath = weightPath;
        }
});

// Permutation
var permutate = (function() {

    var results = [];

    function doPermute(input, output, used, size, level, limit) {
        /*



        */
                if(limit == 1){
                        //var word = output.join('');
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            results.push(output);
            return;
                    }

        if (size == level) {
            //var word = output.join('');
            results.push(output);
            return;
        }

        level++;

        for (var i = 0; i < input.length; i++) {
            if (used[i] === true) {
                continue;
            }
            used[i] = true;
            output.push(input[i]); // pathNode[i]
            doPermute(input, output, used, size, level);
            used[i] = false;
            output.pop();
        }
    }

    return {
        getPermutations: function(input, size, limit) {
            /*

            */
            //var chars = input.split('');
            var output = [];
            var used = new Array(input.length);
            doPermute(input, output, used, size, 0, limit);
            return results;
        }
    };
})();

/*
############################################################
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
##
##        Function
##
#################################################################
*/

Array.prototype.clone = function() {
    var isArr = function(elm){
        return String(elm.constructor).match(/array/i) ? true : false;
    };
    var cloner = function(arr){
        var arr2 = arr.slice(0), len = arr2.length;
        for(var i=0; i < len; i++){
            if( isArr(arr2[i]) )
               arr2[i]=cloner( arr2[i] );
        }
        return arr2;
    };
    return cloner(this);
};


function printRouteToGeneString(route){
        $("#div_genetic").innerHTML = "";
        var seq, pos; // Sequence , Position of Node
        var arrString = new Array();
        for(var i=0; i<=nodeCount; i++){
                arrString[i] = new Array();
        }

        seq = 0;
        for(var i=0; i<route.car.length;i++){
                for(var j=0; j<route.car[i].travels.length;j++){
                seq++;
                        for(var k=0; k<route.car[i].travels[j].path.length;k++){
                                pos = k + 1;
                        arrString[route.car[i].travels[j].path[k]].push("(" + seq + ", " + pos
+")");
                }
        }
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    }

    // print array
    var tmp="";
    for(var i=1; i<arrString.length;i++){ // first string node == 1
                for(var j=0; j<arrString[i].length;j++){
                        if ( i !=1 || j !=0){ // first string node
                                tmp += ", ";
                        }
                        if (j==0){
                                tmp += i + arrString[i][j];
                        }
                        else{
                                tmp += i + "." + j + arrString[i][j];
                        }
        }
    }
    return tmp;
}



function createGeneRoutes(route){
        var limit = 0;
        var seq = 0;
        var tmpPath, tmpWeight;
        var tmp = "";
        var newRoute;
        var pathNodes = new Array();
        var permutePath;
        geneRoutes = new Array();

        // initString
        if(limit < stringLimit){
                newRoute = cloneRoute(route);

                newRoute = calStat(newRoute);
                geneRoutes.push(newRoute);
                limit++;
        }
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
 //$("#div_genetic_string").html("");

/*
//####################

//####################
*/
for(var i=0; i<route.car.length;i++){
	if(limit >= stringLimit){
		break;
	}
	for(var j=0; j<route.car[i].travels.length;j++){
		if(limit >= stringLimit){
			break;
		}
		seq++;
		for(var k=0; k<route.car[i].travels[j].path.length;k++){
			if(limit >= stringLimit){
				break;
			}
			// select path to swap in L index
			for(var l=k+1; l<route.car[i].travels[j].path.length;l++){
				if(limit < stringLimit){
					var newRoute = new Array();
					newRoute = cloneRoute(route);
					// swap
					tmpPath =
newRoute.car[i].travels[j].path[k];
					tmpWeight =
newRoute.car[i].travels[j].weightPath[k];
					newRoute.car[i].travels[j].path[k] =
newRoute.car[i].travels[j].path[l];
					newRoute.car[i].travels[j].weightPath[k]
= newRoute.car[i].travels[j].weightPath[l];
					newRoute.car[i].travels[j].path[l] =
tmpPath;
					newRoute.car[i].travels[j].weightPath[l]
= tmpWeight;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                                        newRoute = calStat(newRoute);


                                        geneRoutes.push(newRoute);
                                        limit++;
                                    }
                                    else{
                                        break;
                                    }
                                } // end for l
                            } // end for k
                        } // end for j
                    } // end for i


                    /*
                        //##########################

                        //##########################
                        */
                        for(var i=0; i<route.car.length;i++){
                            if(limit >= stringLimit){
                                break;
                            }
                            for(var j=0; j<route.car[i].travels.length;j++){
                                if(limit >= stringLimit){
                                    break;
                                }
                                seq++;
                                for(var k=route.car[i].travels[j].path.length-1; k>=0;k--){
                                    if(limit >= stringLimit){
                                        break;
                                    }
                                    // select path to swap in L index
                                    for(var l=k-1; l>=0;l--){
                                        if(limit < stringLimit){
                                            var newRoute = new Array();
                                            newRoute = cloneRoute(route);
                                            // swap
                                            tmpPath =
newRoute.car[i].travels[j].path[k];
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                                                    tmpWeight =
newRoute.car[i].travels[j].weightPath[k];
                                                    newRoute.car[i].travels[j].path[k] =
newRoute.car[i].travels[j].path[l];
                                                    newRoute.car[i].travels[j].weightPath[k]
= newRoute.car[i].travels[j].weightPath[l];
                                                    newRoute.car[i].travels[j].path[l] =
tmpPath;
                                                    newRoute.car[i].travels[j].weightPath[l]
= tmpWeight;


                                                    newRoute = calStat(newRoute);


                                                    geneRoutes.push(newRoute);
                                                    limit++;
                                                }
                                                else{
                                                    break;
                                                }
                                            } // end for l
                                        } // end for k
                                    } // end for j
                                } // end for i

                                /*
                                    //##########################

                                    //##########################
                                    */
                                    var carIndex, travelIndex, pathIndex;
                                    carIndex = 0;
                                    travelIndex = 0;
                                    pathIndex = 0;

                                    while(carIndex<route.car.length){
                                        if(pathIndex>=route.car[carIndex].travels[travelIndex].path.length){
                                            pathIndex = 0;
                                            travelIndex++;
                                        }
```

```
                if(travelIndex>=route.car[carIndex].travels.length){
                        travelIndex = 0;
                        carIndex++;
                }
                if(carIndex>=route.car.length){
                        break;
                }
                for(var i=0; i<route.car.length;i++){
                        if(limit >= stringLimit){
                                break;
                        }
                        for(var j=0; j<route.car[i].travels.length;j++){
                                if(limit >= stringLimit){
                                        break;
                                }
                                seq++;
                                for(var k=0; k<route.car[i].travels[j].path.length;k++){
                                        if(limit >= stringLimit){
                                                break;
                                        }

        if(route.car[carIndex].travels[travelIndex].path.length>=2 &&
route.car[carIndex].travels[travelIndex].path.length<=3){
                                                if(route.car[i].travels[j].path.length>=2
&& route.car[i].travels[j].path.length<=3){
                                                        if(limit < stringLimit){
                                                                var newRoute = new
Array();
                                                                newRoute =
cloneRoute(route);
                                                                // swap
                                                                tmpPath =
newRoute.car[i].travels[j].path[k];
                                                                tmpWeight =
newRoute.car[i].travels[j].weightPath[k];

        newRoute.car[i].travels[j].path[k] =
newRoute.car[carIndex].travels[travelIndex].path[pathIndex];

        newRoute.car[i].travels[j].weightPath[k] =
newRoute.car[carIndex].travels[travelIndex].weightPath[pathIndex];
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            newRoute.car[i].travels[j].path[l] = tmpPath;

            newRoute.car[i].travels[j].weightPath[l] = tmpWeight;



                                                                    newRoute =
calStat(newRoute);



            geneRoutes.push(newRoute);
                                                                    limit++;

                                                            }
                                                            else{
                                                                    break;
                                                            } // end if (limit)

                                                    } // end if
                                            } // end if
                                    } // end for k
                            } // end for j
                    } // end for i
                    pathIndex++;
            }// end while


    return geneRoutes;
}

function cartesian() {
            var r = [], arg = arguments, max = arg.length-1;
            function helper(arr, i) {
                for (var j=0, l=arg[i].length; j<l; j++) {
                    var a = arr.slice(0); // clone arr
                        a.push(arg[i][j]);
                        if (i==max) {
                            r.push(a);
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                } else
                    helper(a, i+1);
                }
        }
    helper([], 0);
    return r;
};


function cloneRoute(route){
        var newRoute = new Array();
        newRoute["car"] = new Array();
        for(var i=0;i<route.car.length;i++){
      newRoute.car[i] = $.extend(true, {}, route.car[i]);
                newRoute.car[i].travels = new Array();

                for(var j=0;j<route.car[i].travels.length;j++){
                        newRoute.car[i].travels[j] = $.extend(true, {},
route.car[i].travels[j]);
                        newRoute.car[i].travels.time = route.car[i].travels.time;
            newRoute.car[i].travels.weight = route.car[i].travels.weight;
            newRoute.car[i].travels.distance = route.car[i].travels.distance;
                        newRoute.car[i].travels[j].path = new Array();
                        newRoute.car[i].travels[j].weightPath = new Array();

                        for(var k=0;k<route.car[i].travels[j].path.length;k++){
                                newRoute.car[i].travels[j].path[k] =
route.car[i].travels[j].path[k];
                                newRoute.car[i].travels[j].weightPath[k] =
route.car[i].travels[j].weightPath[k];
                        }
                }
        }
        //$("#div_genetic_string").html(printRouteToGeneString(newRoute));
        return newRoute;
}


function printAllGeneRoutes(geneRoutes){ // mode 0=defalut(print only selectstring),
1=print all string
        var tmp = "";
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
var table = document.createElement("table");
table.id = "table_string";


var trh = document.createElement("tr");
var header = ["No.", "Gene String","All Weight", "All Distance", "All Time", "All Cost",
"Fitness"];
for (var i = 0; i < header.length; i++) {
    var th = document.createElement("th");
    th.innerText = header[i];
    trh.appendChild(th);
}
table.appendChild(trh);


    for(var i=0;i<geneRoutes.length;i++){

            geneRoutes[i] = calStat(geneRoutes[i]);


var tr = document.createElement("tr");
var td0 = document.createElement("td");
td0.innerText = i+1;
$(td0).css("text-align","left");
$(td0).css("font-size","15px");
tr.appendChild(td0);

var td1 = document.createElement("td");
td1.innerText = printRouteToGeneString(geneRoutes[i]);
$(td1).css("text-align","left");
$(td1).css("font-size","15px");
tr.appendChild(td1);

            var td2 = document.createElement("td");
td2.innerText = geneRoutes[i].allWeight.toFixed(2);
$(td2).css("text-align","left");
$(td2).css("font-size","15px");
tr.appendChild(td2);
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        var td3 = document.createElement("td");
        td3.innerText = geneRoutes[i].allDistance.toFixed(2);
        $(td3).css("text-align","left");
        $(td3).css("font-size","15px");
        tr.appendChild(td3);

        var td4 = document.createElement("td");
        td4.innerText = geneRoutes[i].allTime.toFixed(2);
        $(td4).css("text-align","left");
        $(td4).css("font-size","15px");
        tr.appendChild(td4);

        var td5 = document.createElement("td");
        td5.innerText = geneRoutes[i].allCost.toFixed(2);
        $(td5).css("text-align","left");
        $(td5).css("font-size","15px");
        tr.appendChild(td5);

        var td6 = document.createElement("td");
        td6.innerText = geneRoutes[i].fitness;
        $(td6).css("text-align","left");
        $(td6).css("font-size","15px");
        tr.appendChild(td6);

        table.appendChild(tr);
            }
    return table;
}

/*
 *        สำหรับคำนวนหาค่า weight, distance, time, cost, fitness ของ route
 */

function calStat(route){
            var weight, distance, time, cost;
            var allWeight, allDistance, allTime, allCost, fitness;
            allWeight = allDistance = allTime = allCost = 0;
            fitness = 2;

            for(var i=0; i<route.car.length;i++){
                        //weight = distance = time = cost = 0;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
for(var j=0; j<route.car[i].travels.length;j++){
        weight = distance = time = cost = 0;
        for(var k=0; k<route.car[i].travels[j].path.length;k++){

                weight += route.car[i].travels[j].weightPath[k];


                if(k==0){
                        distance +=
myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k]][0];
                        //log("route.car[i].travels[j].path[k]: " +
route.car[i].travels[j].path[k]);

        //log("myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k]][0] " +
myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k]][0]);

        //log(myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k]][0]);
                }
                if(k>0){
                        if(route.car[i].travels[j].path[k-1] >
route.car[i].travels[j].path[k]){
                                distance +=
myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k-1]][route.car[i].travels[j].path[k]];

        //log(myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k-
1]][route.car[i].travels[j].path[k]]);
                        }
                        else if(route.car[i].travels[j].path[k-1] <
route.car[i].travels[j].path[k]){
                                distance +=
myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k]][route.car[i].travels[j].path[k-1]];

        //log(myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k]][route.car[i].travel
s[j].path[k-1]]);
                        }
                        //log("[][]: " + route.car[i].travels[j].path[k-1] + ",
" + route.car[i].travels[j].path[k]);
                }
                if(k == route.car[i].travels[j].path.length-1){
```

```
                                        distance +=
myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k]][0];

        //log(myMatrix.distanceMatrix.datas[route.car[i].travels[j].path[k]][0]);
                                        }
                                        //log("distance: " + distance );


                                        time += route.car[i].travels[j].weightPath[k] *
myMatrix.timeToLoad;
                                        //log("loadTime: " + time);
                                    } //end for k

                    //log("Distance: " + distance);


                    //time =   (distance / myMatrix.carSpeed) + (weight *
myMatrix.timeToLoad);

                    time +=   ((distance*1000) /  (myMatrix.carSpeed*1000/3600))/60;

                    //log("Time: " + time);

                    cost = myMatrix.getCost(distance);

                    // fitness
                    if(time > myMatrix.limitTime){
                            fitness = 1;
                    }
                    if(weight > myMatrix.carCapacity){
                            fitness = 1;
                    }

                    allWeight += weight;
                    allDistance += distance;
                    allTime += time;
                    allCost += cost;
                    } // end for j


        } // end for i
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
		route.allWeight = allWeight;
		route.allDistance = allDistance;
		route.allTime = allTime;
		route.allCost = allCost;
		route.fitness = fitness;

		return route;
}; // end function calStat


function roulettWheel(geneRoutes){ // รับค่า Parent Gene
		geneRoutes = roulettGeneProb(geneRoutes);
		matingPool = createRoulettRoutes(geneRoutes); parent
		return matingPool;
};

function geneSortFitness(geneRoutes){
		var newGeneRoutes = new Array();
		var fitness2index, fitness1index;
		fitness2index = 0;
		fitness1index = geneRoutes.length -1;
		//selection sort
		for(var i=0; i<geneRoutes.length;i++){
			if(geneRoutes[i].fitness == 2){
				newGeneRoutes[fitness2index]=geneRoutes[i];
				fitness2index++;
			}
			else if(geneRoutes[i].fitness == 1){
				newGeneRoutes[fitness1index]=geneRoutes[i];
				fitness1index--;
			}
		}
		/*
		for(var i=0; i<geneRoutes.length;i++){
			geneRoutes[i] = newGeneRoutes[i];
		}
		*/
		return newGeneRoutes;
};

function roulettGeneProb(geneRoutes){
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
		var sumFitness  = 0;
		// Sum All Fitness
		for(var i=0; i<geneRoutes.length;i++){
			sumFitness += geneRoutes[i].fitness;
		}
		//log(sumFitness);
		var sumProb=0;
		for(var i=0; i<geneRoutes.length;i++){
			geneRoutes[i].p = (geneRoutes[i].fitness/sumFitness); //
geneRoutes[i].p = Probability
			sumProb += geneRoutes[i].p;
			geneRoutes[i].q = (sumProb); // geneRoutes[i].q = Comulative of
Probability
			//log(geneRoutes[i].p + " " + geneRoutes[i].q);
		}
		return geneRoutes;
};

function createRoulettRoutes(geneRoutes){
	var population1, population2;
	var population = new Array();
	var r1, r2;
	population1 = new Array();
	population2 = new Array();
	for(var i=0;i<geneRoutes.length;i++){
		r1 = Math.random(); // random 0-1
		r2 = Math.random(); // random 0-1
		stringNo1 = findRouteFromR(geneRoutes, r1);
		stringNo2 = findRouteFromR(geneRoutes, r2);
		stringSelected = -1;
		population1[i] = geneRoutes[stringNo1];
		population2[i] = geneRoutes[stringNo2];

		//Compare for Select Gene Route

		if(population1[i].fitness > population2[i].fitness){
			population[i] = cloneRoute(population1[i]);
			stringSelected = stringNo1;
		}
		else if(population1[i].fitness < population2[i].fitness){
			population[i] = cloneRoute(population2[i]);
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                    stringSelected = stringNo2;
                }
                else{

                    if(population1[i].allCost < population2[i].allCost){
                        population[i] = cloneRoute(population1[i]);
                        stringSelected = stringNo1;
                    }
                    else if(population1[i].allCost > population2[i].allCost){
                        population[i] = cloneRoute(population2[i]);
                        stringSelected = stringNo2;
                    }
                    else{

                        if(population1[i].allDistance <
population2[i].allDistance){
                            population[i] = cloneRoute(population1[i]);
                            stringSelected = stringNo1;
                        }
                        else if(population1[i].allDistance >
population2[i].allDistance){
                            population[i] = cloneRoute(population2[i]);
                            stringSelected = stringNo2;
                        }
                        else{
                            if(getRandomInt(1,2) == 1){ // min, max
                                population[i] =
cloneRoute(population1[i]);
                                stringSelected = stringNo1;
                            }
                            else{
                                population[i] =
cloneRoute(population2[i]);
                                stringSelected = stringNo2;
                            }
                        } // end if all distance
                    } // end if all Cost
                }//end if all fitness
                /*
                log("pop1:" + r1 + " " + population1[i].q + " " + stringNo1 + " " +
population1[i].fitness);
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                log("pop2:" + r2 + " " + population2[i].q + " " + stringNo2 + " " +
population2[i].fitness);
                log("Select: " + stringSelected);
                log("");
                */
        } // end for loop
        return population;
};

function findRouteFromR(geneRoutes, r){ //search Through Route for r in Cumulative
range
        var found = false;
        index = Math.floor(geneRoutes.length / 2);

        //selection sort
        for(var i=0;i<geneRoutes.length;i++){
                if(geneRoutes[i].q>=r){
                        return i;
                }
        }

        return 0;

};

// Returns a random integer between min and max
// Using Math.round() will give you a non-uniform distribution!
function getRandomInt(min, max) {
  return Math.floor(Math.random() * (max - min + 1) + min);
}



function crossOver(matingPool){
        var crossStringCount;
        var countSelected = 0;
        var route1, route2;
        var r;
        crossStringCount = crossProb * stringLimit;
        crossStringCount = Math.round(crossStringCount);
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
if(crossStringCount%2==1){
	if(crossStringCount<matingPool.length){
		crossStringCount++;
	}
	else if(crossStringCount==matingPool.length){
		crossStringCount--;
	}
}


for(var i=0;i<matingPool.length;i++){
	r = Math.random();
	matingPool[i].r = r;
	if(r<crossProb){
		matingPool.selected = true;
		countSelected++;
	}

}
//
var i=0;
while(countSelected<crossStringCount){
	if(matingPool[i].r>=crossProb){
		r = Math.random();
		matingPool[i].r = r;
		if(r<crossProb){
			matingPool.selected = true;
			countSelected++;
		}
		else{
			matingPool.selected = false;
		}
	}
	i++;
	if(i>=matingPool.length){
		i=0;
	}
}
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
var count=0;
for(var i=0;i<matingPool.length;i++){
	if(matingPool[i].selected == true){
		if(count==0){
			count++;
			route1 = matingPool[i];
		}
		else if(count==1){
			count++;
			route2 = matingPool[i];
		}

		if(count==2){
			count=0;

				var i1, j1, k1, i2, j2, k2, tmp;
				i1 = getRandomInt(0,route1.car.length-1);
				j1 =
getRandomInt(0,route1.car[i1].travels.length-1);
				k1 =
getRandomInt(0,route1.car[i1].travels[j1].path.length-1);
				i2 = getRandomInt(0,route2.car.length-1);
				j2 =
getRandomInt(0,route2.car[i2].travels.length-1);
				k2 =
getRandomInt(0,route2.car[i2].travels[j2].path.length-1);

				//
				tmp = route1.car[i1].travels[j1].path[k1];
				route1.car[i1].travels[j1].path[k1] =
route2.car[i2].travels[j2].path[k2];
				route2.car[i2].travels[j2].path[k2] = tmp;
				//สลับ weightPath
				tmp = route1.car[i1].travels[j1].weightPath[k1];
				route1.car[i1].travels[j1].weightPath[k1] =
route2.car[i2].travels[j2].weightPath[k2];
				route2.car[i2].travels[j2].weightPath[k2] = tmp;
		}
	}
}
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
		return matingPool;
}

function routeToGeneString(route){
		var seq, pos; // Sequence , Position of Node
		var index;
		var arrString = new Array();
		var geneString = new Array();

		for(var i=0; i<=nodeCount; i++){
			arrString[i] = new Array();
		}

		seq = 0;
		for(var i=0; i<route.car.length;i++){
			for(var j=0; j<route.car[i].travels.length;j++){
			seq++;
				for(var k=0; k<route.car[i].travels[j].path.length;k++){
					pos = k + 1;
				//arrString[route.car[i].travels[j].path[k]].push("(" + seq + ", " +
pos +")");
				var node = new Genenode(i, seq, pos,
route.car[i].travels[j].path[k], route.car[i].travels[j].weightPath[k]);
				arrString[route.car[i].travels[j].path[k]].push(node);
			}
		}
	}

	index = 0;
	for(var i=0; i<arrString.length;i++){
			for(var j=0; j<arrString[i].length;j++){
				geneString[index] = arrString[i][j];
			}
		}


	return geneString;
}

/*
 * ################################################################
```

```
 * ###              Mutation
 * ###################################################################
 */
function mutation(matingPool){
        var mutationStringCount;
        var countSelected = 0;
        var route1, route2;
        var r;
        mutationStringCount = mutationProb * stringLimit; //
crossover
        mutationStringCount = Math.round(mutationStringCount);

        //
        if(mutationStringCount%2==1){ //
                if(mutationStringCount<matingPool.length){
                        mutationStringCount++;
                }
                else if(mutationStringCount==matingPool.length){
                        mutationStringCount--;
                }
        }

        //
        for(var i=0;i<matingPool.length;i++){
                r = Math.random();
                matingPool[i].r = r;
                if(r<crossProb){
                        matingPool.selected = true;
                        countSelected++;
                }

        }
        //
        var i=0;
        while(countSelected<mutationStringCount){
                if(matingPool[i].r>=mutationProb){ //
                        r = Math.random();
                        matingPool[i].r = r;
                        if(r<crossProb){
                                matingPool.selected = true;
                                countSelected++;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                        }
                        else{
                                matingPool.selected = false;
                        }
                }
                i++;
                if(i>=matingPool.length){
                        i=0;
                }
        }

        //
        var count=0;
        for(var i=0;i<matingPool.length;i++){
                if(matingPool[i].selected == true){
                        if(count==0){
                                count++;
                                route1 = matingPool[i];
                        }
                        else if(count==1){
                                count++;
                                route2 = matingPool[i];
                        }

                        if(count==2){
                                count=0;
                                //
                                        var i1, j1, k1, i2, j2, k2, tmp;
                                        i1 = getRandomInt(0,route1.car.length-1);
                                        j1 =
getRandomInt(0,route1.car[i1].travels.length-1);
                                        k1 =
getRandomInt(0,route1.car[i1].travels[j1].path.length-1);
                                        i2 = getRandomInt(0,route2.car.length-1);
                                        j2 =
getRandomInt(0,route2.car[i2].travels.length-1);
                                        k2 =
getRandomInt(0,route2.car[i2].travels[j2].path.length-1);

                                        //
                                        tmp = route1.car[i1].travels[j1].path[k1];
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                                        route1.car[i1].travels[j1].path[k1] =
route2.car[i2].travels[j2].path[k2];
                                        route2.car[i2].travels[j2].path[k2] = tmp;
                                        //สลับ weightPath
                                        tmp = route1.car[i1].travels[j1].weightPath[k1];
                                        route1.car[i1].travels[j1].weightPath[k1] =
route2.car[i2].travels[j2].weightPath[k2];
                                        route2.car[i2].travels[j2].weightPath[k2] = tmp;
                                }
                        }
                }
                return matingPool; // offspring
};

function mutationProb(matingPool){
                };

function geneticResult(parentGene, offspringGene){
                var type; //
                //

                var count = 0;
                for(var i=0;i<parentGene.length;i++){
                        parentGene[i].type = 0;
                        //collectGene.push(parentGene[i]);
                        collectGene[count] = parentGene[i];
                        count++;
                }
                for(var i=0;i<offspringGene.length;i++){
                        offspringGene[i].type = 1;
                        //collectGene.push(offspringGene[i]);
                        collectGene[count] = offspringGene[i];
                        count++;
                }
                //
                for(var i=0;i<collectGene.length;i++){
                        collectGene[i] = calStat(collectGene[i]);
                }


                $("#div_genetic_string").append("<br><h1>Collect Gene Unsort: </h1>");
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
		$("#div_genetic_string").append(printAllGeneRoutes(collectGene));

		//
		collectGene = geneSortStat(collectGene);

		return collectGene;

};

function geneSortStat(collectGene){
		var newGeneRoutes = new Array();
		var fitness2index, fitness1index;
		var minCost, minCostIndex;
		var minDistance, minDistanceIndex;
		var lastBestIndex, lastBestIndex2;

		fitness2index = 0;
		fitness1index = collectGene.length -1;
		//Fitness Sort - selection sort
		for(var i=0; i<collectGene.length;i++){
			if(collectGene[i].fitness == 2){
				newGeneRoutes[fitness2index]=collectGene[i];
				lastBestIndex = fitness2index;
				fitness2index++;
			}
			else if(collectGene[i].fitness == 1){
				newGeneRoutes[fitness1index]=collectGene[i];
				fitness1index--;
			}
		}
		log(lastBestIndex);
		collectGene=newGeneRoutes;
		/*
		$("#div_genetic_string").append("<br><h1>Collect Gene Fitness Sort: </h1>");

		$("#div_genetic_string").append(printAllGeneRoutes(collectGene));
		*/

		//Total Cost Sort - selection sort
		newGeneRoutes = new Array();
		lastBestIndex2 = 0;
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
	for(var i=0; i<=lastBestIndex;i++){
		minCost = 999999999;
		for(var j=i; j<=lastBestIndex;j++){
			if(collectGene[j].allCost <= minCost){
				minCost = collectGene[j].allCost;
				minCostIndex = j;
			}
		}
		var tmp = collectGene[i];
		collectGene[i] = collectGene[minCostIndex];
		collectGene[minCostIndex] = tmp;
		if((i>0) && (collectGene[i].allCost == collectGene[0].allCost) &&
(collectGene[i].allCost == collectGene[i-1].allCost) ){
			lastBestIndex2 = i;
		}
	}

	//
	for(var i=0;i<collectGene.length;i++){
		collectGene[i] = calStat(collectGene[i]);
	}

	log(lastBestIndex2);
	//collectGene=newGeneRoutes;
	/*
	$("#div_genetic_string").append("<br><h1>Collect Gene Cost Sort: </h1>");
	$("#div_genetic_string").append(printAllGeneRoutes(collectGene));
	*/

	//Total Distance Sort - selection sort
	lastBestIndex = 0;
	for(var i=0; i<=lastBestIndex2;i++){
		minDistance = 999999999;
		for(var j=i; j<=lastBestIndex2;j++){
			if(collectGene[j].allDistance <= minDistance){
				minDistance = collectGene[j].allDistance;
				minDistanceIndex = j;
			}
		}
		var tmp = collectGene[i];
		collectGene[i] = collectGene[minDistanceIndex];
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                    collectGene[minDistanceIndex] = tmp;
                    if((i>0) && (collectGene[i].allDistance == collectGene[0].allDistance) &&
(collectGene[i].allDistance == collectGene[i-1].allDistance) ){
                        lastBestIndex = i;
                    }
            }

            log(lastBestIndex);
            //collectGene=newGeneRoutes;
            /*
            $("#div_genetic_string").append("<br><h1>Collect Gene Distance Sort: </h1>");

            $("#div_genetic_string").append(printAllGeneRoutes(collectGene));
            */

            geneRoutes = new Array();
            for(var i=0;i<stringLimit;i++){
                    geneRoutes[i] = collectGene[i];
            }
            return geneRoutes;
};

/*
#########################################################
###
###                         FORM ACTION
###
#########################################################
*/

//
$("#input_genetic_btn").click(function () {
            $("#div_genetic_start_btn").hide();
            $("#div_genetic_init_string").html("");
            //
        var table = document.createElement("table");
        table.id = "table_init_string";
        //
        var trh = document.createElement("tr");
        var header = ["Initial String"];
        for (var i = 0; i < header.length; i++) {
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        var th = document.createElement("th");
        th.innerText = header[i];
        trh.appendChild(th);
    }
    table.appendChild(trh);

    var tr = document.createElement("tr");
    var std0 = document.createElement("td");

    std0.innerText = printRouteToGeneString(cloneRoute(initRoute));
    $(std0).css("text-align","left");
    $(std0).css("font-size","15px");
    tr.appendChild(std0);
    table.appendChild(tr);
    $("#div_genetic_init_string").append(table);
    $("#div_genetic_init_string").show();

    // set variable
    stringLimit =  $("#input_num_init_string").val();
    crossProb = $("#input_crossover_prob").val();
                mutationProb= $("#input_mutation_prob").val();
                geneticLoop = $("#input_num_loop").val();
                //log("genetic Loop: " + geneticLoop);

                // Create Initial Genes
                createGeneRoutes(initRoute); // permutation
                $("#div_genetic_string").html("");
    $("#div_genetic_string").append(printAllGeneRoutes(geneRoutes));
        $("#div_genetic_string").show();
                parentGene = geneSortFitness(geneRoutes);


                // Create Offspring
                for(var i=0;i<geneticLoop;i++){
                        var no = i+1;
                        matingPool = roulettWheel(parentGene);
                        //$("#div_genetic_string").append("<br><h1>roulettWheel
Loop:" + no+ "</h1>");

        //$("#div_genetic_string").append(printAllGeneRoutes(matingPool));
```

```
                matingPool = crossOver(matingPool);
                //$("#div_genetic_string").append("<br><h1>CrossOver Loop:" +
no+ "</h1>");

        //$("#div_genetic_string").append(printAllGeneRoutes(matingPool));

                offspringGene = mutation(matingPool);
                //$("#div_genetic_string").append("<br><h1>Mutation (Offspring)
Loop:" + no+ "</h1>");

        //$("#div_genetic_string").append(printAllGeneRoutes(matingPool));

                parentGene = geneticResult(parentGene, offspringGene);

                $("#div_genetic_string").append("<br><h1>Result Loop:" + no+
"</h1>");

        $("#div_genetic_string").append(printAllGeneRoutes(parentGene));

                for(var j=0;j<parentGene.length;j++){
                        parentGene[j] = calStat(parentGene[j]);
                }
                parentGene = geneSortFitness(parentGene);
            }
});
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียนวิทยานิพนธ์

นายนพรุจ สังข์แป้น เกิดเมื่อวันที่ 31 ธันวาคม 2531 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนเซนต์คาเบรียล จังหวัดกรุงเทพมหานคร และสำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมการผลิต จากภาควิชาวิศวกรรมการผลิต คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ เมื่อปีการศึกษา 2553 และได้เข้ารับการศึกษาต่อในหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2554

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University