อภินันทนาการ

i.3150094

การใช้อัลกอลิทึมพันธุกรรมแก้ปัญหาด้านการเดินทางของพนักงานขาย

Using genetic algorithm for traveling salesman problem

นายปรเมศวร์ ธนารุณ
นายปริญญา เกษร
นายเรวัฒน์ ธีระแนว

สำนักหอสมุด มหาวิทยาลัยนเรศวร
วันลงทะเบียน 22 ก.ค. 2547
เลขทะเบียน 4740403
เลขเรียกหนังสือ QA 76 .9 .D35 ป1710 2545

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2545

ใบรับรองโครงการวิศวกรรมอุตสาหการ

หัวข้อโครงงานวิศวกรรมอุตสาหการ	: การใช้อัลกอลิทึมพันธุกรรมแก้ปัญหาด้านการเดินทาง ของพนังงานขาย
ผู้ดำเนินงานวิศวกรรมอุตสาหการ	: นายปรเมศวร์ ธนารุณ รหัส 42361089 นายปริญญา เกษร รหัส 42361097 นายเรวัฒน์ ธีระแนว รหัส 42361329
ที่ปรึกษาโครงงานวิศวกรรมอุตสาหการ	: ดร.ภูพงษ์ พงษ์เจริญ
สาขาวิชา	: วิศวกรรมอุตสาหการ
ภาควิชา	: วิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา	: 2545

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานวิศวกรรมอุตสาหการฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมอุตสาหการ

คณะกรรมการสอบโครงงานวิศวกรรมอุตสาหการ

.....ประธานกรรมการ
(ดร.ภูพงษ์ พงษ์เจริญ)

.....กรรมการ
(ผศ.ดร. กวิน สนธิเพิ่มพูน)

.....กรรมการ
(อาจารย์ศรีสัจจา บุญฤทธิ์)

.....กรรมการ
(อาจารย์โพธิ์งาม รัตนโชติ)

หัวข้อโครงงานวิจัย	: การใช้โปรแกรมกระบวนการเชิงพันธุกรรมแก้ปัญหาด้านการเดินทางของพนักงานขาย : กรณีศึกษาภายในประเทศไทย
ผู้ดำเนินงานวิจัย	: นายปรเมศวร์ ธนารุณ รหัส 42361089
	: นายปริญญา เกษร รหัส 42361097
	: นายเรวัฒน์ ธีระแนว รหัส 42361329
ที่ปรึกษาโครงงานวิจัย	: ดร.ภูพงษ์ พงษ์เจริญ
ระดับการวิจัย	: วิศวกรรมศาสตร์บัณฑิต วศ.บ. (วิศวกรรมอุตสาหการ)
สาขาวิชา	: วิศวกรรมอุตสาหการ
ภาควิชา	: วิศวกรรมอุตสาหการ
ปีการศึกษา	: 2545

.....

บทคัดย่อ

ปัญหาการเดินทางในประไทยเป็นปัญหาที่น่าสนใจมากปัญหาหนึ่งในหลายๆปัญหาที่มีปัญหาการเดินทางนี้เป็นที่รู้จักกันในชื่อการหาค่าเหมาะสมในการรวมลำดับ ปัญหานี้ถูกทำให้มีความยุ่งยากจากจำนวนเมืองต่างๆและลำดับการเลือก นักเดินทางบางคนอาจจะเลือกที่จะเดินทางไปเฉพาะเมืองใหญ่ๆถ้าพวกเขามีเวลาจำกัด และบางคนก็เลือกที่จะเดินทางไปในอีกหลายๆเมืองเมื่อพวกเขาไม่ต้องถูกจำกัดด้วยเวลา

กระบวนการเชิงพันธุกรรมเป็นกระบวนการความรู้วิธีหนึ่งที่สามารถแก้ปัญหาการหาค่าเหมาะสมในการรวมลำดับ งานวิจัยนี้มีจุดมุ่งหมายเพื่อสร้างโปรแกรมที่ใช้แก้ปัญหาการเดินทางโดยใช้กระบวนการเชิงพันธุกรรม โดยโปรแกรมนี้ได้สร้างระบบการติต่อกับผู้ใช้เป็นแบบรูปภาพหรือวัตถุซึ่งผู้ใช้สามารถกำหนดตัวแปรเชิงพันธุกรรมได้ตามความต้องการ และงานวิจัยนี้นี้ได้ทำการทดสอบค่าตัวแปรเชิงพันธุกรรมแบบอนุกรม ซึ่งได้ทำการแล้วว่าโปรแกรมมีความสามารถในการหาคำตอบที่ดีขึ้นเมื่อมีการให้กำเนิดมากขึ้นและได้คำตอบว่าการตั้งค่าประชากรสูงๆจะทำให้มีโอกาสที่จะได้คำตอบที่ดีมากขึ้น โดยข้อมูลที่ใช้เป็นระยะทางระหว่าง 76 จังหวัดเป็นค่าจริงที่ได้อ้างอิงจากกรมทางหลวงแผ่นดิน

Project Title	: Using genetic algorithm for traveling salesman problem	
	: A case study in Thailand	
Project Developer	: Dhanarun P.	Student code 42361089
	: Kesorn P.	Student code 42361097
	: Theeranaew R.	Student code 42361329
Project Advisor	: Pongcharoen P. (Dr.)	
Level of study	: Bachelor Degree of Engineering (Industrial Engineering)	
Major	: Industrial Engineering	
Department	: Industrial Engineering	
Academic Year	: 2002	

Abstract

Traveling Problem especially in Thailand is one of the interesting problem ,the problem is know as combinatorial optimization problem. The problem is involved with the sequence of visiting is a number of cities. Some traveler may prefer to visit some large cities if they have time limitation . Some may prefer to visit more cities when there is no time constrain have.

Genetic Algorithm is one of well know algorithm to solve combinatorial optimization problem. This project aims to develop a program to solve traveling problem using Genetic Algorithm . The program is design with friendly graphic user interface which allow user to select Genetic parameter as user wish. The program has been test with a series of experiment. The result show that it can obtain better solution (shorter distance) when increasing number of generation and especially number of population is increased. And this project using real distance of 76 Thailand's province by Department of State Highways.

กิตติกรรมประกาศ

ในการทำโครงงานวิศวกรรมครั้งนี้ คณะผู้จัดทำได้รับความอนุเคราะห์จากบุคคลหลายฝ่ายทำให้สามารถทำงานได้สำเร็จลุล่วงได้ด้วยดี รวมถึงเพื่อนๆที่คอยให้กำลังใจในยามที่อ่อนล้า และคนใกล้ชิดทั้งหมดที่ให้ความเข้าใจและคอยดูแลในยามที่ทุ่มเทกับงาน ทางคณะผู้จัดทำจึงขอขอบคุณมา ณ ที่นี้

คณะผู้จัดทำขอกราบขอบพระคุณ ดร.ภูพงษ์ พงษ์เจริญ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานตามที่ท่านได้ให้ความกรุณา อบรมประสิทธิ์ประสาทความรู้เกี่ยวกับการทำงาน อีกทั้งยังสละเวลาตรวจสอบงานอย่างสม่ำเสมอ และยิ่งต้องขอบคุณเป็นพิเศษสำหรับคำแนะนำและแนวทางแก้ปัญหา จนทำให้ทุกปัญหาเป็นเพียงสิ่งที่ผ่านมาและผ่านไป

นายปรเมศวร์ ธนารุณ
นายปริญญา เกษร
นายเรวัฒน์ ธีระแนว

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ง

บทคัดย่อภาษาอังกฤษ.....จ

กิตติกรรมประกาศ.....ฉ

สารบัญตาราง.....ญ

สารบัญภาพ.....ฎ

บทที่ 1 บทนำ..... 1

1.1 หัวข้อโครงงาน.....1

1.2 ชื่ออาจารย์ที่ปรึกษา.....1

1.3 หลักการ ทฤษฎี เหตุผล หรือสมมุติฐาน.....1

1.4 วัตถุประสงค์ของการศึกษา..... 1

1.5 ประโยชน์ที่ได้รับจากการศึกษา..... 1

1.6 ขอบเขตและวิธีการวิจัย.....1

บทที่ 2 กระบวนการเชิงพันธุกรรม(Genetic Algorithm)..... 2

2.1 อะไรคือ Genetic algorithm (GA).....2

2.2 ความแข็งแรงของกระบวนการ Optimization และกระบวนการค้นหาโดยทั่วไป...3

2.3 กระบวนการการหาคำตอบที่ดีที่สุด ด้วยการเปลี่ยนแปลงทางพันธุกรรม GA..... 3

2.5 การค้นหาโดยใช้พื้นฐานของ (Population-Based Search)..... 8

2.7 การถอดรหัสปัญหา (Encoding problem)..... 9

2.8 การสรรหา (Selection)..... 12

2.9 Genetic algorithm แบบผสม (Hybrid genetic algorithm).....15

2.10 ความแตกต่างของประสิทธิภาพในแต่ละเทคนิค Optimization 16

2.11 การใช้ Genetic Algorithm (GA)..... 17

2.12 การใช้กระบวนการ Tabu Search (TS).....19

2.14 การใช้กระบวนการ Simulated annealing (SA).....32

2.15 การใช้กระบวนการ Neural Network..... 33

2.16 บทความเกี่ยวกับการประยุกต์ใช้ Genetic algorithm การแก้ปัญหาต่างๆ.....34

สารบัญ(ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินงาน 36

3.1 ศึกษาหลักการและทฤษฎีของ Genetic Algorithm(GA) 36

3.2 ศึกษาปัจจัยต่างๆ ที่เกี่ยวข้องกับกระบวนการ 36

3.3 แผนภาพการทำงานของ GA 38

3.4 ขั้นตอนการทำงานของ Genetic Algorithm 39

3.9 กระบวนการทำงานที่ใช้เป็น FUNCTION ภายในโปรแกรม 44

3.10 ศึกษาการเขียนโปรแกรมด้วย Visual Basic 47

3.11 การออกแบบโปรแกรม 49

3.12 การเขียนโปรแกรมและทดสอบ 53

3.13 การพัฒนาและแก้ไขโปรแกรม 54

3.14 สมมติฐานและขอบเขตของโปรแกรม 54

บทที่ 4 การวิจัยและผลของการวิจัย 56

4.1 การทดสอบการทำงานของโปรแกรม 56

4.2 ผลการทดสอบ 57

4.3 การทดลองเพื่อหาการกำหนดค่าที่เหมาะสมภายใต้ข้อจำกัด 89

4.4 การทดลองเกี่ยวกับฐานของการคัดสรร Selection Base 93

4.5 สรุปการทำงานทั้งหมดของ โปรแกรม 98

บทที่ 5 สรุปผลการดำเนินงานวิจัย 99

5.1 สรุปผลการดำเนินงานวิจัย 99

5.2 ข้อแนะนำ 99

หนังสืออ้างอิง 102

ภาคผนวก ก. 103

การติดตั้งและถอนการติดตั้งโปรแกรม 103

วิธีการใช้งานโปรแกรม 108

สารบัญ(ต่อ)

หน้า

ภาคผนวก ข. 117
แผนภาพขั้นตอนและวิธีการดำเนินงานวิจัย.....117

ประวัติผู้วิจัย.....118

สารบัญตาราง

ตารางที่ หน้า

2.1 การหาพื้นที่ของคำตอบและการแก้ปัญหา....17
2.2 ค่าที่ได้จากกระบวนการหาคำตอบสำหรับฟังก์ชัน อย่างไม่ต่อเนื่อง....28
2.3 ค่าที่ได้จากกระบวนการหาคำตอบสำหรับฟังก์ชัน อย่างไม่ต่อเนื่อง....29
2.4 ค่าที่ได้จากกระบวนการหาคำตอบสำหรับฟังก์ชัน อย่างไม่ต่อเนื่อง....29
2.5 ค่าที่ได้จากกระบวนการหาคำตอบสำหรับฟังก์ชัน อย่างไม่ต่อเนื่อง....30
2.6 ระยะทางในการเดินทางที่ได้รับ....34
4.1 ข้อมูลตั้งแต่ GENERATION ที่ 1 ถึง 100 (SEED 69)....60
4.2 ข้อมูลตั้งแต่ GENERATION ที่ 1 ถึง 100 (SEED 2)....66
4.3 ข้อมูลตั้งแต่ GENERATION ที่ 1 ถึง 100 (SEED 90)....74
4.4 ข้อมูลตั้งแต่ GENERATION ที่ 1 ถึง 100 (SEED 12)....81
4.5 ข้อมูลของเครื่อง COMPUTER ที่ใช้ในการทดสอบ....90
4.6 ผลการทดลอง....90
4.7 ภาพชุดเปรียบเทียบผลลัพธ์จากการทดลองการกำหนดค่าที่เหมาะสมภายใต้ข้อจำกัด....91
4.8 SPEC เครื่องที่ใช้เทียบอ้างอิงเวลา....92
ก.1 ความต้องการขั้นต่ำของระบบ....103

สารบัญภาพ

ภาพที่ หน้า

2.1 โครงสร้างโดยทั่วไปของ Genetic algorithm.....6

2.2 การเปรียบเทียบระหว่างกระบวนการหาคำตอบโดยทั่วไปกับ (GA)..... 8

2.3 พื้นที่ของรหัสและพื้นที่ของคำตอบ..... 10

2.4 การเปลี่ยนแปลงเคลื่อนย้ายจากโครโมโซมไปสู่คำตอบ..... 11

2.5 ประสิทธิภาพการคัดสรรจากจำนวนประชากรที่เท่าเดิม..... 13

2.6 ประสิทธิภาพการคัดสรรจากจำนวนประชากรที่มากขึ้น.....14

2.7 แสดงการใช้กระบวนการทาง Genetic algorithm (GA)..... 18

2.8 - 2.23 ฮิสโตแกรมของค่าฟังก์ชันจากกระบวนการต่างๆ..... 20 - 28

2.24 แผนที่ของเมืองทั้ง 50 เมือง..... 31

3.1 แผนภาพการทำงานของ GA..... 38

3.2 population..... 39

3.3 การกำเนิดลูกจาก กระบวนการ Crossover Operation.....40

3.4 One-point crossover (Murata & Ishibuchi,1994).....40

3.5 การกำเนิดลูกจาก กระบวนการ Mutation Operation.....41

3.6 การกำเนิดลูกจาก กระบวนการ Inversion mutation (Goldberg,1989)..... 41

3.7 Fitness measure.....42

3.8 Roulette Wheel..... 42

3.9 Decoding.....43

3.10 One-point crossover (Murata & Ishibuchi,1994).....44

3.11 Position based crossover (Syswerda,1991).....44

3.12 Two-point centre crossover (Murata & Ishibuchi,1994)..... 45

3.13 Centre inverse mutation (Tralle,2000).....45

3.14 Inversion mutation (Goldberg,1989)..... 45

3.15 Two-operation adjacent swap (Murata & Ishibuchi,1994).....46

3.16 Three-operation adjacent swap (Murata & ishibuchi,1994)..... 46

3.17 Two-operation random swap (Murata & Ishibuchi,1994)..... 46

สารบัญภาพ (ต่อ)

ภาพที่ หน้า

3.18 Three-operation random swap (Murata & Ishibuchi,1994)..... 47

3.19 Shift operation mutation (Murata & Ishibuchi,1994)..... 47

3.20 – 3.26 แผนภูมิแสดงขั้นตอนการดำเนินงานของโปรแกรม..... 49 - 52

4.1 กราฟแสดงความเปลี่ยนแปลงใน 100 GENERATION..... 63

4.2 กราฟแสดงความเปลี่ยนแปลงใน 100 GENERATION..... 70

4.3 กราฟแสดงความเปลี่ยนแปลงใน 100 GENERATION..... 72

4.4 กราฟแสดงความเปลี่ยนแปลงใน 100 GENERATION..... 84

4.5 กราฟของระยะทางการเดินทางเฉพาะภาคเหนือ..... 85

4.6 แผนภาพแสดงเส้นทางเดินภายในภาคเหนือจากคำตอบที่ 1ST GENERATION..... 86

4.7 แผนภาพแสดงเส้นทางเดินภายในภาคเหนือจากคำตอบที่ 14TH GENERATION..... 87

4.8 แผนภาพแสดงเส้นทางเดินภายในภาคเหนือจากคำตอบที่ 59TH GENERATION..... 88

4.9 รูปเปรียบเทียบเส้นทางเดินทั้ง 3 GENERATION..... 89

4.10 กราฟเปรียบเทียบผลลัพธ์การกำหนดฐานของการคัดสรร(SEED 2)..... 94

4.11 กราฟเปรียบเทียบผลลัพธ์การกำหนดฐานของการคัดสรร(SEED 12)..... 95

4.12 กราฟเปรียบเทียบผลลัพธ์การกำหนดฐานของการคัดสรร(SEED 69)..... 96

4.13 กราฟเปรียบเทียบผลลัพธ์การกำหนดฐานของการคัดสรร(SEED 90)..... 97

ก.1 – ก.8 เริ่มติดตั้งโปรแกรม..... 103 -106

ก.9 การเรียกใช้งาน..... 107

ก.10 – ก.12 การถอนการติดตั้ง..... 107 -108

ก.13 – ก.24 วิธีการใช้งานของโปรแกรม..... 109 -116

ข.1 แผนภาพแสดงวิธีและขั้นตอนการดำเนินงานวิจัย..... 117

บทที่ 1
บทนำ

1.1 หัวข้อโครงงาน

การใช้โปรแกรมกระบวนการเชิงพันธุกรรมแก้ปัญหาด้านการเดินทาง ของพนักงานขาย : กรณีศึกษาภายในประเทศไทย

1.2 ชื่ออาจารย์ที่ปรึกษา

ดร. ภูพงษ์ พงษ์เจริญ

1.3 หลักการ ทฤษฎี เหตุผล หรือสมมุติฐาน

การศึกษาในปัจจุบันนี้ได้ใช้ทฤษฎีความรู้ที่ใช้มานานแล้ว แต่ปัจจุบันนี้มีผู้ค้นคว้าหาวิธีการใหม่ๆมากมายแต่ในหลักสูตรการศึกษาทั่วไปไม่ได้มีการเรียนการสอนในจุดนี้ จึงทำให้ผู้จัดทำเกิดความสนใจและทำการศึกษาและนำเสนอออกมาในรูปแบบของโปรแกรมเพื่อช่วยในการใช้งานและเรียนรู้ได้ง่ายขึ้น

1.4 วัตถุประสงค์ของการศึกษา

1.4.1 เพื่อเขียนโปรแกรม Intelligence Optimization โดยใช้ Genetic Algorithm ให้สามารถใช้งานได้ง่าย

1.4.2 ศึกษาหาวิธีการใช้งานพารามิเตอร์บางตัวเพื่อให้ได้คำตอบที่เหมาะสมและประหยัดเวลา

1.5 ประโยชน์ที่ได้รับจากการศึกษา

1. สามารถนำความรู้ในเชิงทฤษฎีใหม่ๆมาประยุกต์ใช้ในทางปฏิบัติได้จริง
2. สามารถนำมาเป็นตัวช่วยในการวิเคราะห์และวางแผนการบริหารงาน
3. เป็นการนำความรู้ใหม่มานำเสนอในรูปแบบของโปรแกรมคอมพิวเตอร์

1.6 ขอบเขตและวิธีการวิจัย

ศึกษาวิธีการคิดแบบ Genetic แล้วนำมาสร้างโปรแกรมคอมพิวเตอร์โดยใช้โปรแกรม Visual Basic 6.0 ให้ใช้งานได้ง่ายและมีประสิทธิภาพ

บทที่ 2

กระบวนการเชิงพันธุกรรม (Genetic algorithm Optimization)

2.1 อะไรคือ Genetic algorithm (GA)

Genetic algorithm (GA) เป็นกระบวนการค้นหาที่มีพื้นฐานมาจากกระบวนการเลือกสรรและพันธุกรรมทางธรรมชาติโดยจะเป็นการรวมของโครงสร้างแบบสายที่แข็งแรงที่สุดที่สามารถอยู่รอดได้ด้วยการแลกเปลี่ยนข้อมูลโครงสร้างด้วยวิธีการสุ่มเพื่อจะกำหนดวิธีการค้นหาด้วยสติปัญญาของมนุษย์ ในทุก ๆรุ่นจะมีการสร้างโครงสร้างแบบสายขึ้นมาใหม่โดยใช้โครงสร้างแบบสายที่แข็งแรงที่สุดจากรุ่นเก่าและรุ่นใหม่ที่สร้างขึ้น GA จะมีระบบการสุ่มโดยที่จะอาศัยข้อมูลที่มีอยู่แล้วเพื่อหาผลงานที่ดียิ่งขึ้น

Genetic algorithm (GA) ได้รับการพัฒนาขึ้นโดย John Holland เพื่อนร่วมงานและนักเรียนที่ University of Michigan จุดประสงค์ของการทำการวิจัยนี้มีด้วยกัน 2 ประการคือ

1. เพื่ออธิบายการปรับตัวของระบบธรรมชาติ

2. เพื่อออกแบบและประดิษฐ์โปรแกรมระบบที่นำความสำคัญของกระบวนการของระบบธรรมชาติมาใช้ ซึ่งจากจุดประสงค์ข้อที่ 2 นี้ทำให้มีการค้นพบที่สำคัญทั้งในทางระบบธรรมชาติและสิ่งที่ประดิษฐ์ขึ้นอีกมากมาย

หัวข้อเรื่องที่เป็นที่สนใจในการทำวิจัยเกี่ยวกับ Genetic algorithm (GA) คือ ความมีสุขภาพดี (robustness) นั่นคือ ความสมดุลระหว่างความมีประสิทธิภาพและความจำเป็นในการที่จะอยู่รอดในสิ่งแวดล้อมหลายๆแบบ ซึ่งความมีสุขภาพดีของ Genetic algorithm (GA) สามารถนำไปประยุกต์ใช้ได้หลากหลายในระบบที่สร้างขึ้นซึ่งถ้าสามารถปรับปรุงความมีสุขภาพดีนี้ได้หลายวิธีก็จะสามารถช่วยลดราคาต้นทุนในการออกแบบได้ รวมทั้งยังสามารถทำให้ระบบมีระยะเวลาในการใช้งานที่นานขึ้นและมีประสิทธิภาพที่ดีขึ้นด้วยแต่ผู้ออกแบบและประดิษฐ์ระบบนี้สามารถปรับปรุงระบบได้ที่ความมีสุขภาพดี และความมีประสิทธิภาพเท่านั้นส่วนการซ่อมแซมด้วยตัวเอง การกำหนดแนวการดำเนินการและแพร่พันธุ์จะเป็นการกำหนดโดยกระบวนการทางธรรมชาติซึ่งหาได้ยากในระบบที่ซับซ้อนอันอื่น

จากที่กล่าวมาด้านบนสามารถสรุปได้ว่าการศึกษาระบบการทำงานที่ดีควรมีการเรียนรู้จากระบบธรรมชาติ Genetic algorithm (GA) ได้รับการพิสูจน์ทางทฤษฎี และจากประสบการณ์แล้วว่าสามารถทำการค้นหาได้ดีในกลุ่มตัวอย่างที่ซับซ้อนได้ บทความที่เกี่ยวกับเรื่องนี้โดยตรงเป็นของ

Holland (1975) Adaptation in Natural and Artificial systems มีบทความและวิทยานิพนธ์หลายฉบับที่นำเอาเทคนิคนี้ไปใช้ในกระบวนการ Optimization และการควบคุม (Control) เนื่องจากประสิทธิภาพของ Genetic algorithm (GA) ปัจจุบันพบว่ามีการนำเอา Genetic algorithm(GA) ไปใช้อย่างแพร่หลายในทางธุรกิจ วิทยาศาสตร์และวิศวกรรมซึ่งเหตุผลหลักเนื่องมาจากความง่ายแต่มีประสิทธิภาพในการคำนวณของ Genetic algorithm (GA) รวมทั้งการค้นหาของGenetic algorithm(GA)ไม่ได้ถูกควบคุมด้วยการตั้งสมมุติฐานของพื้นที่ตัวอย่าง

2.2 ความแข็งแรงของกระบวนการ Optimization และกระบวนการค้นหาโดยทั่วไป

งานวิจัยในปัจจุบันมีการบ่งชี้ว่ามีกระบวนค้นหา 3 ชนิดได้แก่

1. การค้นหาด้วยพื้นฐานทางแคลคูลัส (Calculus-based Methods)
2. การค้นหาแบบการสังเกต (Enumerative Scheme)
3. การค้นหาแบบสุ่ม (Random Search Algorithm)

กระบวนการค้นหาด้วยพื้นฐานทางแคลคูลัสได้มีการค้นคว้าเกี่ยวกับวิธีนี้อย่างแพร่หลายซึ่งสามารถแบ่งได้เป็น 2 ขั้นใหญ่ ๆนั่นคือ ทางตรงและทางอ้อม กระบวนการทางอ้อมจะเป็นกระบวนการค้นหา Local Optimum ด้วยการแก้ปัญหาชุดสมการแบบไม่เป็นเชิงเส้น ซึ่งเป็นผลมาจากการตั้งค่าอนุพันธ์ของสมการเท่ากับศูนย์ซึ่งวิธีการนี้เป็นการวางนัยทั่วไปแบบหลายมิติ ของแคลคูลัสเบื้องต้นในการหาจุดที่ดีที่สุด ถ้ามีการให้ฟังก์ชันต่อเนื่องที่ไม่มีข้อจำกัด แล้วการหาจุดที่ดีที่สุดจะเริ่มหาจากที่มีความชันเป็น0ในทุกทิศทางในทางตรงกันข้ามการค้นหาทางตรงเป็นการค้นหา Local Optimum ด้วยการเคลื่อนไหวหาคำตอบในทิศทางที่เกี่ยวข้องกับอนุพันธ์ในช่วงใกล้ ๆ กัน ซึ่งเรียกง่ายๆว่า Bill Climbing เพื่อที่จะหาจุดที่ดีที่สุดในช่วงใกล้ๆกัน แล้วกระโดดไปหาฟังก์ชันที่มีทิศทางปรับปรุงค่าคำตอบให้ดีที่สุด

2.3 กระบวนการการหาคำตอบที่ดีที่สุด ด้วงการเปลี่ยนแปลงทางพันธุกรรม (Genetic algorithm optimization)

ปัจจุบันมีการนำกระบวนการการหาคำตอบที่ดีที่สุดเท่าที่จะหาได้ (Optimization) ไปประยุกต์ใช้ในการแก้ปัญหาหลายด้านไม่ว่าจะเป็นทางด้าน Scheduling หรือการหาระยะทางสั้นที่สุดในการเดินทาง

1. มีการพัฒนาวิธีของ LP, Dynamic Programming, B&B
2. มีการพัฒนาวิธี Simulated annealing(SA), Tabu search(TS) และ GA

3. เป็นการแก้ปัญหาที่อาศัยกระบวนการเฟ้นสุ่มทางสถิติ (Stochastic Process) ในการหาค่าตอบที่ดีในสุดในพื้นที่ตัวอย่าง

4. ไม่สามารถที่จะหาค่าตามในพื้นที่ตัวอย่างทั้งหมดได้ ไม่การันตี Optimal Solution

Genetic algorithm (GA) เป็นกระบวนการเฟ้นสุ่มทางสถิติในการประมาณค่าคำตอบ ที่ดีที่สุดของพื้นที่ตัวอย่างที่ซับซ้อน โดยในการทำงานของ Genetic Algorithm (GA) จะใช้การเปรียบเทียบกับวิวัฒนาการทางชีววิทยาโดยคิดว่าความสมบูรณ์ของร่างกายคนเป็นสิ่งสำคัญในการอยู่รอดและสืบพันธุ์ กระบวนการ Genetic Algorithm (GA) เริ่มจากการถอดปัญหาเพื่อสร้างรายการของยีน หลังจากนั้นยีนจะรวมกันอย่างสุ่มเพื่อสร้างเป็นโครโมโซมซึ่งแต่ละโครโมโซมแสดงถึง

ปัญหาในทางด้านอุตสาหกรรม เป็นปัญหาที่ซับซ้อนและยากที่จะแก้ไขด้วยวิธีการ Optimization แบบพื้นฐาน ตั้งแต่ปี 1960 มีความสนใจมากขึ้นที่จะทำการเลียนแบบธรรมชาติเพื่อนำมาแก้ไขปัญหาเหล่านั้น

การเลียนแบบวิวัฒนาการทางธรรมชาติทำให้เกิดเทคนิคการแก้ปัญหาแบบผันเปลี่ยนตามความเหมาะสม เรียกว่า Evolutionary algorithms ซึ่งวิธีการนี้จะให้ผลที่ดีกว่าวิธีการแก้ปัญหาพื้นฐาน เมื่อนำไปประยุกต์ใช้กับปัญหาที่เกิดขึ้นในโลก ปัจจุบันการวิจัยจะแบ่งกระบวนการเหล่านี้เป็น 3 อย่าง Genetic algorithms (GAs) , Evolutionary programming (EP) และSolution Strategle (ESs) ซึ่งระหว่าง 3 กระบวนการนี้ Genetic algorithm (GA) เป็นวิธีที่เป็นที่รู้จักมากที่สุด

ในปัจจุบันนี้ Genetic Algorithm (GA) เป็นที่ได้รับความสนในมากเพราะ ความสามารถของเทคนิคของการแก้ปัญหาที่ซับซ้อนรวมทั้งนำไปประยุกต์ใช้ อย่างแพร่หลายในด้านวิศวกรรมอุตสาหการ ได้แก่ Scheduling and Sequening การออกแบบความน่าเชื่อถือ , การวางผังการเดินรถ , เทคโนโลยีแบบกลุ่มการวางผังโรงงานอื่น ๆ

2.3.1 โครงสร้างทั่วไปของ Genetic algorithm (GA) โครงสร้าง Genetic algorithm (GA) ได้รับการพัฒนาโดยนาย Goldberg Genetic algorithm(GA) เป็นเทคนิคการค้นหาแบบผันเปลี่ยนโดยอาศัยพื้นฐานกลไกการเลือกและการสืบพันธุ์ตามธรรมชาติซึ่ง Genetic algorithm (GA) จะแตกต่างจากเทคนิคโดยทั่ว ๆไป Genetic algorithm (GA) เริ่มจากการตั้งกลุ่มเป้าหมายแบบสุ่มเรียกว่า ประชากรซึ่งประชากรแต่ละคนจะถูกเรียกว่าโครโมโซม โครโมโซมมีโครงสร้างแบบสายโดยทั่วไปจะเป็น Binary Bits โครโมโซมจะมีวิวัฒนาการตามจำนวนรอบที่เกิดขึ้นใหม่เรียกว่า รุ่น (Generation) แต่ละรุ่นจะถูกประเมินผลโดยใช้การวัดความแข็งแรง สมบูรณ์ของโครโมโซม ในการที่จะสร้างรุ่นใหม่ ซึ่งเรียกว่า Oftspring จะถูกสร้างโดยการรวมของโครโมโซม จากรุ่นของปัจจุบัน หรือการคัดประชากรทิ้งเพื่อให้เหลือจำนวนของกลุ่มประชากรเท่าเดิมโครโมโซม หลังจากวิวัฒนาการหลาย ๆรุ่นกระบวนการจะเข้าไปสู่โครโมโซมนี้ดีที่สุด ซึ่งโครโมโซมจะแสดงถึง

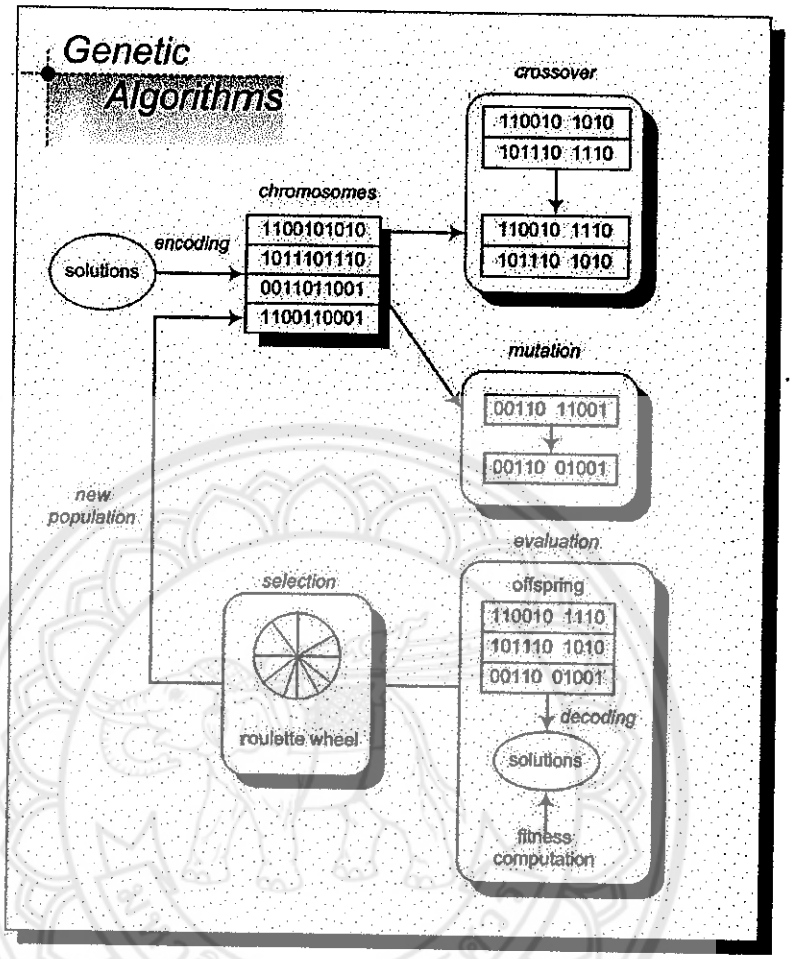
คำตอบที่เหมาะสมที่สุดเมื่อให้ P(t) และ C(t) แสดงถึงรุ่นพ่อแม่ไปในรุ่น t C(t) แสดงถึงลูกในรุ่น t โครงสร้างโดยทั่วไปของ GA ดังแสดงใน

Precedure : Genetic Algorithms

```
begin
  t ← 0
  initialize P( t )
  evalution P( t )
while ( not termination condition ) do
  recombine P( t ) to yield C( t )
  evalution C( t ) :
  select P( t+1 ) from P( t ) and C( t )
  t ← ( t+1 )
  end
end
```

กระบวนการด้านบนเป็นภาพที่แบบปรับปรุงของ Baker's [192,287]GA ประกอบด้วย 2 กระบวนการหลัก ๆ

1. Genetic operation : การตัดสลับและการกลายพันธุ์
2. Evolution operation การสรรหา
3. Genetic operation เป็นการเลียนแบบกระบวนการสร้างประชากรใหม่ในแต่ละรุ่น ส่วนการสรรหา (Selection) เป็นการเลียนแบบวิวัฒนาการตามแนวคิดของดาร์วิน



ภาพที่ 2.1 โครงสร้างโดยทั่วไปของ Genetic algorithm

(ที่มา : Runwei Chenga*, Mitsuo Genb, Yasuhiro Tsujimurab ,A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies, ,Computers & Industrial Engineering 36 (1999) 343±364 หน้า 3)

การตัดสลับ (Crossover) เป็นกระบวนการหลักทางพันธุกรรมซึ่งเกิดจากการรวมตัวกันของ 2 โครโมโซมที่ทำให้ลูกเกิดมานั้นมีลักษณะผสม วิธีการตัดสลับนี้ง่ายที่สุด คือการสุ่มหาตำแหน่งของจุดตัดแล้วสุ่มเลือกโครโมโซมจากแต่ละด้านของจุดตัดแล้วนำมาผสมกัน

ซึ่งกระบวนการนี้จะทำงานได้ดีกับสายพันธุ์โครงสร้างแบบบิต ประสิทธิภาพของ Genetic algorithm (GA) ขึ้นอยู่กับวิธีการที่ใช้ในการตัดสลับ

อัตราการตัดสลับ (Crossover rate) คืออัตราของลูกในแต่ละรุ่นเมื่อเทียบกับประชากรทั้งหมด (Population size) อัตราส่วนนี้จะเป็นตัวบอกถึงค่าเฉลี่ยของการตัดสลับ อัตราการตัดสลับที่สูงจะทำให้ขอบเขตของคำตอบที่เป็นไปได้กว้างขึ้น และลดอัตราการเจอคำตอบที่ผิดพลาดแต่ถ้าค่าอัตราการตัดสลับสูงมากเกินไปเวลาที่ใช้ในการหาคำตอบก็จะมากขึ้นไปด้วย

การกลายพันธุ์ (Mutation) จะทำให้เกิดการเปลี่ยนแปลงอย่างต่อเนื่องในหลาย ๆ โครโมโซม วิธีการที่ง่ายในการกลายพันธุ์ คือการเปลี่ยนแปลงยีนในโครโมโซมใน
การกลายพันธุ์มีบทบาทที่สำคัญ

1. ในเรื่องของการทดแทนยีนที่หายไปในระหว่างกระบวนการสรรหา

2. การค้นหายีนที่ไม่ปรากฏในจำนวนประชากรเริ่มต้น (Initial population) ชั้นแรก

อัตราการกลายพันธุ์ (Mutation rate) คือเปอร์เซ็นต์ของจำนวนยีนทั้งหมดใน จำนวนประชากรทั้งหมด อัตราการกลายพันธุ์เป็นตัวควบคุมอัตราการเกิดของยีนใหม่ใน จำนวนประชากรนั้น ๆ ถ้าเกิดอัตรานี้มีค่าน้อยมากเกินไป จะทำให้ยีนที่มีประโยชน์ไม่ได้ถูกนำมาใช้ ถ้าอัตรานี้มีค่ามากเกินไป จะทำให้กระบวนการสุ่มมีความยุ่งยากมากขึ้น ซึ่งมีผลทำให้รุ่นลูกมีการเลียนแบบรุ่นพ่อแม่น้อยลง และกระบวนการจะสูญเสียความสามารถในการเรียนรู้จากการค้นหาที่ผ่านมา

2.4 การสำรวจและการนำไปใช้ประโยชน์

การค้นหาเป็นหนึ่งในวิธีการแก้ปัญหาที่ใช้อย่างแพร่หลายในปัญหาที่ไม่สามารถกำหนดจุดเริ่มต้นของคำตอบได้ การค้นหาสามารถกระทำได้ด้วยกลยุทธ์ ไม่มีแนวทางและมีแนวทางกลยุทธ์แบบไม่มีแนวทางจะไม่มีการใช้ข้อมูลจากขอบเขตของปัญหา ถ้าหากลยุทธ์แบบมีแนวทางจะนำข้อมูลมาใช้เพิ่มเติมประกอบการพิจารณาหาแนวทางค้นหาที่ดีที่สุด หัวเรื่องสำคัญ 2 ประการ ในกลยุทธ์การค้นหาคือ

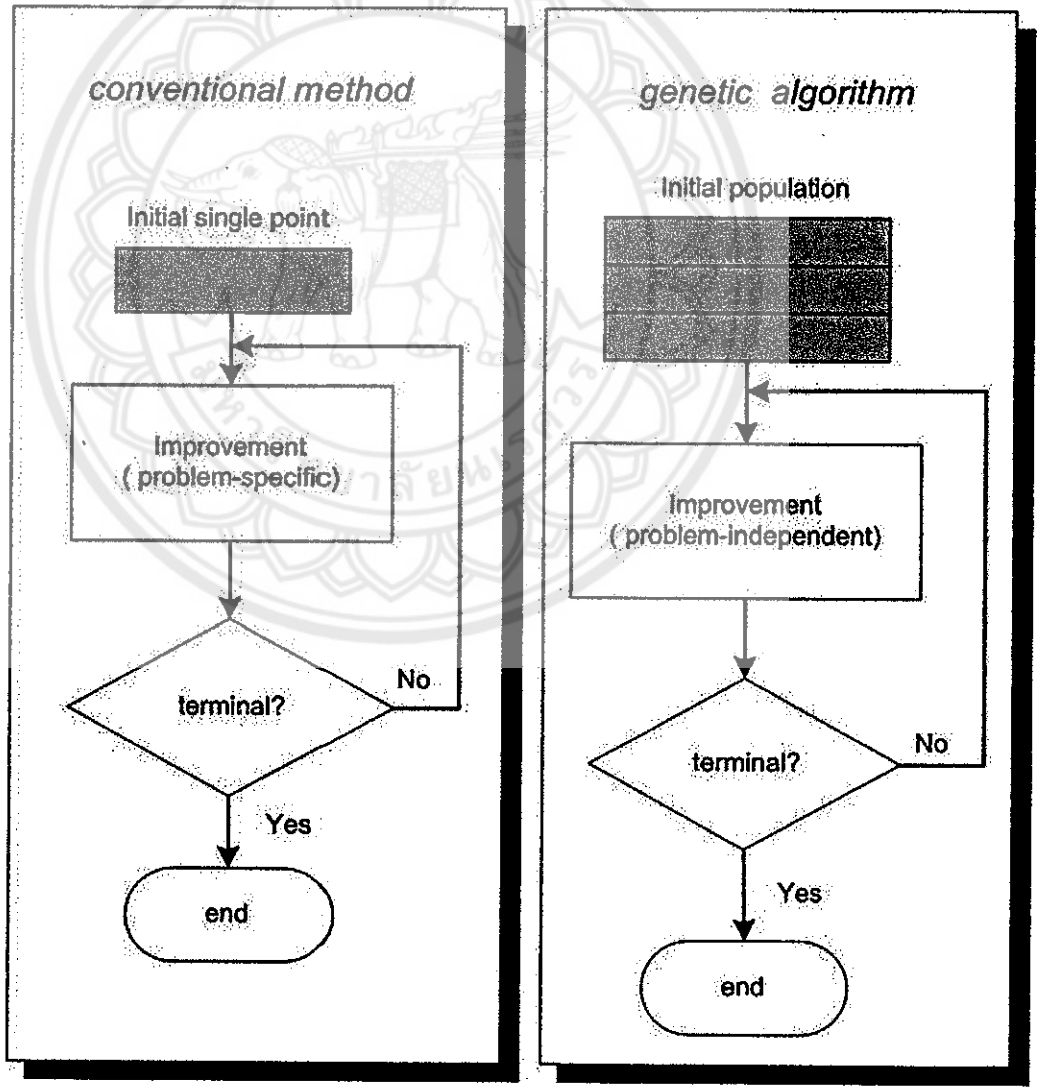
1. การนำคำตอบที่ดีที่สุดไปใช้ประโยชน์

2. มีการสำรวจพื้นที่ค้นหา

Michalewicz ได้ทำการเปรียบเทียบ การค้นหาแบบปีนภูเขา การค้นหาแบบสุ่ม และการค้นหาตามพันธุกรรม การค้นหาแบบปีนภูเขาเป็นการค้นทำที่ดีในการนำคำตอบที่ดีที่สุดไปใช้ประโยชน์ เพื่อการปรับปรุงคำตอบโดยไม่สนใจการสำรวจพื้นที่ค้นหา Genetic algorithm (GA) เป็นวิธีการค้นหาแบบไม่เฉพาะเจาะจง ซึ่งรวมเอาการค้นหาแบบมีแนวทางและการค้นหาแบบผันแปร ทำให้เกิดความสมดุลในการนำคำตอบที่ดีที่สุดไปใช้ประโยชน์และการสำรวจพื้นที่ค้นหาได้ในช่วงเริ่มต้นของการประยุกต์ใช้วิธีการหาแบบพันธุกรรม จำนวนประชากรมีหลากหลายและการตัดสลับเป็นที่นำไปใช้ได้อย่างกว้างขวางในการสำรวจพื้นที่ค้นหาเมื่อมีการพัฒนาคำตอบที่แข็งแรงที่สุด การตัดสลับจะทำให้เกิดการสำรวจสิ่งแวดล้อม แต่ไม่ใช้การสำรวจตัวเองนอกจากนั้นแล้ววิธีการสำรวจทางพันธุกรรมแบบพื้นฐานเป็นการออกแบบเพื่อใช้ในกระบวนการค้นหาทั่ว ๆไป ซึ่งเป็นกระบวนการค้นหาแบบไม่มีทิศทางและผลของการค้นหาไม่รับประกันว่าจะได้สิ่งที่ดีขึ้น

2.5 การค้นหาโดยใช้พื้นฐานของ (Population-Based Search)

กระบวนการการหาคำตอบที่ดีที่สุดเท่าที่จะหาได้ (Optimization) แบบทั่ว ๆ ไปจะลำดับการคำนวณโดยมีพื้นฐานมาจากอนุพันธ์ของ Function อุปสงค์ วิธีการนี้นำไปใช้กับจุดจุดเดียวในพื้นที่ค้นหา (ภาพที่ 2.2) เสี่ยงมากที่จะนำพาไปสู่คำตอบที่เป็นคำตอบที่ดีที่สุดเท่าที่จุดจุดหนึ่ง(Local optima) Genetic algorithm GA จะทำคำตอบในหลาย ๆด้านโดยคงจำนวนของ จำนวนประชากรที่น่าจะให้คำตอบที่ดีที่สุดไว้ ซึ่งวิธีการนี้จะช่วยให้หลีกพ้นจากคำตอบที่ดีที่สุดที่จุดๆหนึ่ง (Local optima) ได้ ในแต่ละรุ่นของจำนวนประชากร คำตอบที่ดีจะถูกสร้างขึ้น ในขณะที่คำตอบที่ไม่ดีจะถูกทำลายไป Genetic algorithm (GA) จะใช้กฎการเปลี่ยนแปลงแบบความน่าจะเป็นเพื่อที่จะเลือกบุคคลในการเจริญพันธุ์หรือสูญพันธุ์ ซึ่งจะนำไปสู่ผลคำตอบที่น่าจะดีขึ้น



ภาพที่2.2 การเปรียบเทียบระหว่างกระบวนการหาคำตอบโดยทั่วไปกับ Genetic algorithm (GA)

2.6 ข้อได้เปรียบหลักของ Genetic algorithm (GA)

3 ข้อได้เปรียบหลักเมื่อมีการประยุกต์ Genetic algorithm (GA) เข้าไปใช้ในกระบวนการ Optimization

1. Genetic algorithm (GA) ไม่มีข้อจำกัดทางคณิตศาสตร์ในการแก้ปัญหา Optimization มากขึ้น เนื่องจากการใช้วิวัฒนาการทางธรรมชาติ การหาคำตอบของ Genetic algorithm (GA) จะเป็นไปโดยไม่คิดถึงภาพที่แบบของปัญหา Genetic algorithm (GA) สามารถแก้ไขปัญหาได้ในทุกภาพที่แบบของวัตถุประสงค์ ข้อจำกัดต่อเนื่องไม่ต่อเนื่องหรือพื้นที่ค้นหาแบบผสม

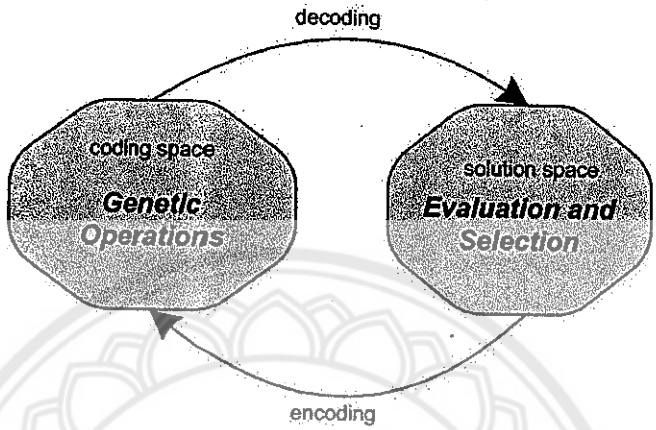
2. วิวัฒนาการของตัวการทำให้ Genetic algorithm (GA) สามารถค้นหาในพื้นที่การค้นหาทั้งหมดได้ซึ่งจะไม่เหมือนแบบการค้นหาทั่วไปที่เป็นการค้นหาแบบจุดต่อจุด อาศัยกระบวนการเบนเข้าหากันแบบขั้น

3. Genetic algorithm (GA) มีความยืดหยุ่นสูงสามารถนำไปประยุกต์ใช้ได้กับหลาย ๆปัญหาที่มี ลักษณะเฉพาะเจาะจง

2.7 การถอดรหัสปัญหา (Encoding problem)

การถอดรหัสปัญหาเข้าสู่ภาพที่แบบของโครโมโซมอย่างไร เป็นหลักสำคัญสำหรับกระบวนการทางพันธุกรรม ในการวิจัยของ Holland การถอดรหัสของปัญหา การถอดรหัสทำด้วยโครงสร้างสายเส้นคู่ สำหรับการนำ Genetic algorithm (GA) ไปประยุกต์ใช้ เป็นการถอดรหัสโดยใช้โครงสร้างสายเส้นคู่ทำได้ไม่ง่ายนัก ดังนั้นจึงมีการพัฒนาเทคนิคการถอดรหัสแบบไม่เป็นสายเส้นเพื่อใช้ในปัญหาบางประเภท เช่น การถอดรหัสเลขจำนวนจริงในการแก้ปัญหา Optimization แบบไม่มีข้อจำกัด การเลือกคำตอบที่เหมาะสมเมื่อเป็นตัวแทนในการตอบปัญหา เป็นพื้นฐานสำคัญในการประยุกต์ใช้ Genetic algorithm (GA) ในการแก้ปัญหาจริง

ลักษณะสำคัญหนึ่งอย่างของ Genetic algorithm (GA) คือ การทำงานบนพื้นที่รหัสและพื้นที่คำตอบสลับกันไป โดย Genetic operations จะทำงานบนพื้นที่รหัส ในขณะที่กระบวนการประเมินผลและการสรรหา(Evaluation and Selection) จะทำงานบนพื้นที่คำตอบ ดังแสดงในภาพที่2.3



ภาพที่ 2.3 พื้นที่ของรหัสและพื้นที่ของคำตอบ

(ที่มา : Runwei Chenga*, Mitsuo Genb, Yasuhiro Tsujimurab ,A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies, ,Computers & Industrial Engineering 36 (1999) 343±364 หน้า 18)

การสรรหาทางธรรมชาติ เป็นความสัมพันธ์ระหว่างโครโมโซมและประสิทธิภาพในการถอดรหัสของคำตอบสำหรับกระบวนการโครงสร้างการใส่รหัสแบบไม่เป็นเส้น สิ่งสำคัญ 3 ประการที่เกิดขึ้นในกระบวนการนี้ระหว่างโครโมโซมและคำตอบ คือ ความเป็นไปได้ของโครโมโซม

1. ความถูกต้องตามแบบแผนของโครโมโซม
2. ความเป็นเอกลักษณ์ในการใส่และถอดรหัส

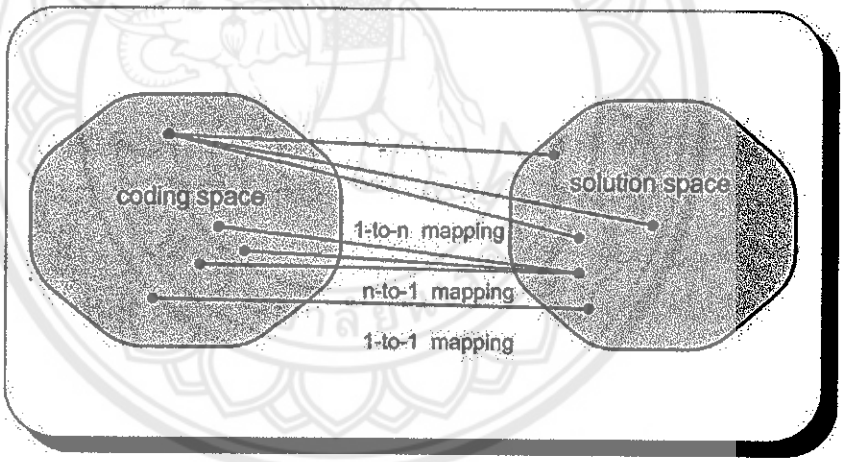
ความเป็นไปได้ หมายถึง การที่รหัสที่ถอดจากคำตอบจากโครโมโซมนั้น อยู่บริเวณที่เป็นไปได้ของปัญหา ความถูกต้องตามแบบแผนกล่าวถึง การที่โครโมโซมจะแสดงถึงคำตอบของปัญหา ดังแสดงในภาพที่ 2.3

ความเป็นไปไม่ได้ของโครโมโซม เกิดขึ้นเนื่องจากธรรมชาติของปัญหา Optimization แบบมีข้อจำกัดนั่นเอง ในปัญหาเพื่อหาคำตอบที่ดีที่สุดเท่าที่จะทำได้(Optimization) แบบจำกัด ค่าคำตอบที่ดีที่สุดมักจะเกิดขึ้นบริเวณริมของพื้นที่ที่เป็นไปได้และเป็นไปไม่ได้ กระบวนการลงโทษจะทำให้การค้นหาทางพันธุกรรมไปหาคำตอบที่ดีที่สุดทั้งทางด้านที่เป็นไปได้และเป็นไปไม่ได้

ความไม่ตรงตามแบบแผนของโครโมโซม เกิดขึ้นเนื่องจากการถอดรหัสสำหรับการแก้ปัญหา Optimization การถอดรหัสถูกนำมาใช้ซึ่งอาจทำให้เกิดภาพที่ที่ไม่เป็นไปตามแบบแผนจากกระบวนการตัดสลับ โดยเลือกกระทำ ณ จุดๆหนึ่ง เนื่องจากโครโมโซมที่ไม่ถูกต้องตามแบบแผน ไม่สามารถนำมาถอดรหัสได้ ซึ่งแสดงว่าโครโมโซมนั้นไม่สามารถนำมาแทนค่าได้ ดังนั้นกระบวนการลงโทษจึงถูกนำมาใช้ เทคนิคการซ่อมแซมจะถูกสร้างขึ้นมาเพื่อจะเปลี่ยนโครโมโซมที่ไม่เป็นไปตามแบบแผน วิธีการที่รู้จักกันดี คือ การนำ PMX ซึ่งเป็นการตัดสลับที่จุด 2 จุดในการแก้ปัญหานี้ การใส่และถอดรหัสจากโครโมโซมไปสู่คำตอบ จะประกอบไปด้วย 1 ใน 3 กรณีเหล่านี้

1. การใส่และถอดรหัส 1 ต่อ 1
2. การใส่และถอดรหัสจาก N ไป 1
3. การใส่และถอดรหัสจาก 1 ไป N

ดังแสดงในภาพที่ 2.4



ภาพที่ 2.4 การเปลี่ยนแปลงเคลื่อนย้ายจากโครโมโซมไปสู่คำตอบ

(ที่มา : Runwei Chenga*, Mitsuo Genb, Yasuhiro Tsujimurab ,A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies, ,Computers & Industrial Engineering 36 (1999) 343±364 หน้า 18)

การใส่และถอดรหัส 1ต่อ 1 เป็นวิธีที่ดีที่สุด แต่การใส่รหัสแบบ 1 ไป N เป็นวิธีที่ไม่ต้องการที่สุด

2.8 การสรรหา (Selection)

Genetic algorithm (GA) ได้ประยุกต์นำเอาการสร้รหาตามธรรมชาติของ Darvin การสรรหาทำให้เกิดแรงผลักดันใน Genetic algorithm (GA) โดยทั่วไปแล้วแรงกดดันในการสรรหาขั้นต่ำเป็นตัวชี้ให้เห็นถึงจุดเริ่มต้นของกระบวนการค้นหาแบบ Genetic algorithm (GA) ในการสำรวจขอบเขตปัญหาขนาดใหญ่ ในขณะตรงกันข้าม แรงกดดันในการสรรหาขั้นสูง ควรจะนำไปใช้ในตอนจบกระบวนการ เพื่อจะหาคำตอบที่ดีที่สุดในขอบเขตปัญหานั้นๆ การสรรหาเป็นการกำหนดแนวทางให้ Genetic algorithm (GA) ดำเนินการไปในพื้นที่ที่น่าจะให้คำตอบที่ดีขึ้น ซึ่งในการสรรหานั้น ประกอบด้วยส่วนสำคัญ 3 ส่วน คือ

2.8.1 พื้นที่ในการกำหนดตัวอย่าง (Sampling space)

2.8.2 กระบวนการในการกำหนดตัวอย่าง (Sampling mechanism)

2.8.3 ความน่าจะเป็นในการสรรหา (Selection probability)

2.8.1 พื้นที่ในการกำหนดตัวอย่าง (Sampling space) กระบวนการสรรหาสามารถทำให้เกิดประชากรรุ่นใหม่ โดยจะเกิดจากพ่อแม่และลูกทั้งหมดหรือบางส่วน ซึ่งนำไปสู่ปัญหาในการกำหนดพื้นที่ตัวอย่าง จะถูกตัดสินด้วย 2 ปัจจัย คือ ขนาดและส่วนประกอบกำหนดให้

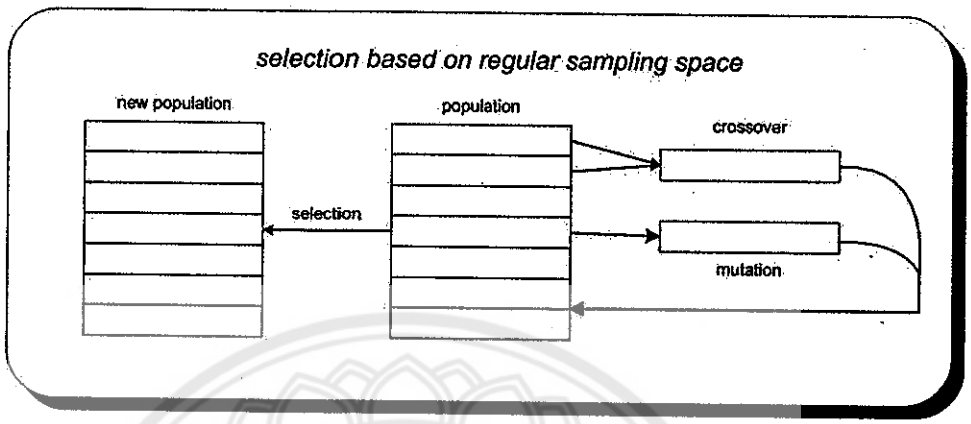
pop_size คือขนาดของประชากร

off_size คือขนาดของลูกที่เกิดขึ้นในแต่ละรุ่น

พื้นที่ในการกำหนดตัวอย่างแบบทั่วไป(Regular sampling space)จะมีขนาดเท่ากับขนาดของ pop_Size ซึ่งจะประกอบไปด้วยลูกทั้งหมดรวมถึงบางส่วนของพ่อแม่เท่านั้น พื้นที่ในการกำหนดตัวอย่างแบบขยาย (Enlarged sampling) จะมีขนาดเท่ากับขนาด pop_size e + off_size รวมถึงพ่อแม่และลูกทั้งหมด

พื้นที่ในการกำหนดตัวอย่างแบบทั่วไป(Regular sampling space) พ่อแม่จะถูกทดแทนโดยลูกของตัวเองหลังจากการให้กำเนิดโดยทันทีซึ่งจะเรียกกระบวนการนี้ว่า " การแทนที่รุ่น (Generation replacement) " เนื่องจากวิธีการทางกรรมพันธุ์นี้ เป็นวิธีที่ไม่มีแนวทางตามธรรมชาติ ลูกที่ถูกผลิตขึ้นมาอาจมีคุณสมบัติที่ด้อยกว่าพ่อแม่ได้ ด้วยการแทนที่รุ่นนี้ทำให้บางโครโมโซมที่มีความสมบูรณ์มากกว่าอาจหายไปในวิวัฒนาการได้ Michalewicz ได้ให้คำอธิบายในกระบวนการ Genetic algorithm (GA) แบบพื้นฐาน ซึ่งลูกในแต่ละรุ่นจะเข้ามาแทนที่พ่อแม่ในทันทีหลังจากเกิดรุ่นต่อไปจะต้องผ่านการสรรหาแบบ Roulette wheel selection ดังแสดง

ในภาพที่ 2.5

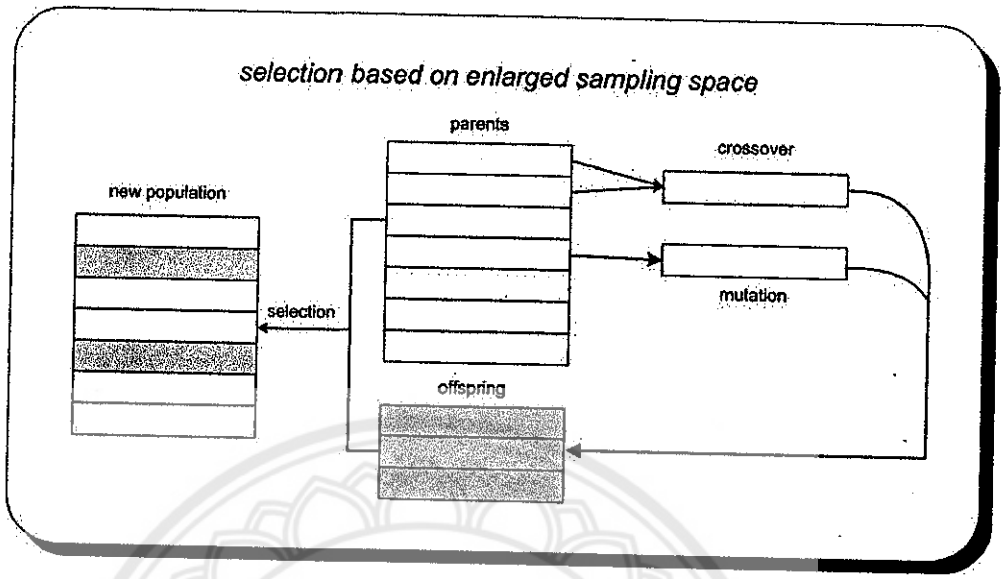


ภาพที่ 2.5 ประสิทธิภาพการคัดสรรจากจำนวนประชากรที่เท่าเดิม

(ที่มา : Runwei Chenga*, Mitsuo Genb, Yasuhiro Tsujimurab ,A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies, ,Computers & Industrial Engineering 36 (1999) 343±364 หน้า 21)

พื้นที่ในการกำหนดตัวอย่างแบบขยาย (Enlarged sampling) เมื่อการสรรหากระทำบนพื้นที่ขนาดใหญ่ ทั้งพ่อแม่และลูกมีโอกาสเท่ากันในการแข่งขันเพื่อความอยู่รอด ส่วนมากเรียกวิธีการนี้ว่า " การสรรหาแบบ $(\mu+\lambda)$ " ด้วยวิธีการนี้ พ่อแม่ μ และลูก λ แข่งขันเพื่อนการอยู่รอดซึ่งกันและกัน μ ที่ดีที่สุดจากพ่อแม่รุ่นเก่าจะถูกคัดเลือกให้เป็นพ่อแม่รุ่นต่อไป ในอีกกรณีหนึ่ง

กลยุทธ์การวิวัฒนาการ คือ การสรรหาแบบ (μ,λ) ซึ่ง μ ที่ดีที่สุดจากลูก จะเป็นพ่อแม่รุ่นต่อไป $(\mu<\lambda)$ ดังแสดงในภาพที่ 2.6



ภาพที่ 2.6 ประสิทธิภาพการคัดสรรจากจำนวนประชากรที่มากขึ้น

(ที่มา : Runwei Chenga*, Mitsuo Genb, Yasuhiro Tsujimurab ,A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies, ,Computers & Industrial Engineering 36 (1999) 343±364 หน้า 22)

ข้อได้เปรียบของกระบวนการนี้ คือ ความสามารถในการเพิ่มประสิทธิภาพของ GA ด้วยการเพิ่มอัตราการตัดสลับและอัตราการกลายพันธุ์

2.8.2 กระบวนการในการกำหนดตัวอย่าง (Sampling mechanism) เป็นปัญหาของการคัดเลือกโครโมโซมอย่างไรจากพื้นที่ตัวอย่างนั้นๆ กระบวนการพื้นที่ 3 ประการ ในการคัดเลือกโครโมโซม ประกอบด้วย

2.8.2.1 การสุ่มตัวอย่างแบบไม่แน่นอน (Stochastic sampling)

2.8.2.2 การสุ่มตัวอย่างแบบแน่นอน (Deterministic sampling)

2.8.2.3 การสุ่มตัวอย่างแบบผสม (Mixed sampling)

2.8.2.1 การสุ่มตัวอย่างแบบไม่แน่นอน(Stochastic sampling) ลักษณะทั่วไปของวิธีการนี้ คือ การเลือกโดยนำจำนวนของโครโมโซมทั้งหมดมาใช้ประกอบการตัดสินใจโดยอาศัยความน่าจะเป็นในการอยู่รอด ซึ่งวิธีการนี้ประกอบด้วย 2 ส่วน คือ

1. การหาค่าเฉลี่ยของจำนวนโครโมโซม
2. การแปลงค่าเฉลี่ยนั้นให้เป็นจำนวนของลูก

ค่าเฉลี่ยของจำนวนโครโมโซมเป็นจำนวนจริงซึ่งแสดงให้เห็นถึงจำนวนเฉลี่ยของลูกที่

โครโมโซมหนึ่งๆควรจะมี กระบวนการเลือกจะถูกนำมาใช้ในการแปลงค่าจริงเหล่านั้นให้เป็นจำนวนของลูก

2.8.2.2 การสุ่มตัวอย่างแบบแน่นอน(Deterministic sampling) กระบวนการนี้จะเป็นการเลือกโครโมโซมที่ pop_Size ที่ดีที่สุดจากขอบเขตตัวอย่างทั้งการสรรหาแบบ $\mu+\lambda$ และ μ,λ รวมอยู่ในวิธีการนี้ด้วย ทั้ง 2 กระบวนการข้างต้น ได้มีการป้องกันการนำเข้าของโครโมโซมในการเพิ่มจำนวนประชากรระหว่างการสรรหา ดังนั้นวิธีการนี้จึงเป็นวิธีการที่นักวิจัยนิยมนำมาใช้ในการแก้ปัญหา

2.8.2.3 การสุ่มตัวอย่างแบบผสม(Mixed sampling) กระบวนการนี้จะเป็นการรวมทั้งแบบลักษณะแบบสุ่มและแบบแน่นอนไปพร้อมๆกัน ตัวอย่าง เช่น การสรรหาทัวร์ โดย Goldberg วิธีการนี้จะสุ่มเลือกเซตของ โครโมโซมและเลือกตัวที่ดีที่สุดเพื่อนำมาทำการสืบพันธุ์ต่อไป

2.8.3 ความน่าจะเป็นในการสรรหา (Selection probability)

จะเป็นการตัดสินใจเกี่ยวกับการกำหนดความน่าจะเป็นในการสรรหาสำหรับโครโมโซมแต่ละตัว ซึ่งความน่าจะเป็นนี้ขึ้นอยู่กับความสมบูรณ์ของโครโมโซมนั่นเอง ในรุ่นแรกๆแนวโน้มโครโมโซมที่ดีมากบางตัวจะเป็นตัวที่เด่นในการเลือกแต่ละรุ่นต่อมา เมื่อประชากรมีขนาดใหญ่เพียงพอการแข่งขันระหว่าง โครโมโซมจะลดน้อยลง ซึ่งจะเป็นผลทำให้เกิดการค้นหาแบบสุ่ม

กลไกการตั้งค่าและเรียงลำดับ ได้นำมาประยุกต์ใช้ในการตั้งค่าจะเป็นการกำหนดค่าของฟังก์ชั่นเป้าหมายและค่าความน่าจะเป็นเพื่อการอยู่รอดสำหรับแต่ละ โครโมโซม ถูกกำหนดขึ้นมาจากค่าฟังก์ชั่นเป้าหมายนั้นๆ ส่วนการจัดลำดับ จะใช้การจัดลำดับของโครโมโซมในการตัดสินความน่าจะเป็นในการอยู่รอด ซึ่งการตั้งค่าความแข็งแรงของโครโมโซมนี้ เนื่องมาจาก เหตุผล 2 ประการ คือ

1. เพื่อรักษาระดับความแตกต่างของการเรียงลำดับความแข็งแรงของโครโมโซมที่เหมาะสมเอาไว้
2. เพื่อป้องกันการถูกครอบงำด้วยตัวโครโมโซมที่แข็งแรงมากกว่าในช่วงแรกของการแข่งขันและเพื่อกระตุ้นให้เกิดการแข่งขันในตอนหลัง

2.9 Genetic algorithm แบบผสม (Hybrid genetic algorithm)

Genetic algorithm (GA) เป็นวิธีการที่พิสูจน์แล้วว่าในการแก้ไขปัญหา Optimization ได้อย่างหลากหลายและ มีประสิทธิภาพ แต่อย่างไรก็ตามในหลายๆสถานการณ์ Genetic algorithm (GA) แบบพื้นฐานไม่สามารถแก้ไขได้ ดังนั้น กระบวนการในการผสมวิธีหลายๆอย่างมารวมกันจึงได้ถูก

นำมาใช้ Genetic algorithm(GA) แบบผสมที่เป็นที่รู้จักกันมากอย่างหนึ่ง คือ การนำเอา Optimization ในบริเวณนั้นๆเข้ามาเป็นวงเพิ่มเติมสำหรับการรวมและการสรรหา(Selection)ใน กระบวนการ Genetic algorithm (GA) แบบพื้นฐาน ด้วยวิธีการผสมนี้ Optimization บริเวณนั้นจะ ถูกนำมาใช้ทุกครั้ง เมื่อมีการสร้างลูกรุ่นใหม่เพื่อที่จะย้ายรุ่นลูกนี้ไปสู่จุดที่ดีที่สุดดังกล่าวก่อนที่จะ รวมไปเป็นประชากร Genetic algorithm(GA) จะนำไปใช้ในการสำรวจหาประชากรทั้งหมด ส่วน วิธีการแบบที่มีแนวทางจะถูกนำไปใช้ในการค้นหาจุดที่ดีที่สุดในบริเวณรอบๆ โครโมโซม โดย ส่วนมากแล้วกระบวนการแบบผสมนี้ จะให้ประสิทธิภาพที่ดีกว่าในการนำแต่ละวิธีไปปฏิบัติโดย ไม่มีการผสม การวิจัยที่ผ่านมามีการค้นพบว่ากลไกทางธรรมชาติได้มีการนำไปประยุกต์ใช้ใน กระบวนการแบบผสมเหล่านี้ ตัวอย่างได้แก่ วิวัฒนาการตามแนวคิดของ Lamarckin และ กระบวนการ Menetic

2.10 ความแตกต่างของประสิทธิภาพในแต่ละเทคนิค Optimization โดยการทดสอบด้วยฟังก์ชั่น Benchmark

ฟังก์ชันในการทดสอบทั้ง 4 ฟังก์ชัน [De Jong, 1975] ได้ถูกนำไปใช้สำหรับการทดสอบถึง ประสิทธภาพของกระบวนการการหาคำตอบสำหรับคำตอบที่เป็นตัวเลขที่ดีที่สุดเท่าที่จะทำได้ ฟังก์ชั่นทั้ง 4 ฟังก์ชันมีดังนี้

$$F_1 = \sum_{i=1}^{3} x_i^2 \tag{2.1}$$

$$F_2 = 100\left(x_1^2 - x_2\right)^2 + \left(1 - x_1\right)^2 \tag{2.2}$$

$$F_3 = \sum_{i=1}^{5} \left(x_i\right) \tag{2.3}$$

$\left(x_i\right)$ จะต้องมีค่าน้อยกว่าหรือเท่ากับ x_i

$$F_4 = \left[0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} \left(x_i - a_{ij}\right)^6} \right]^{-1} \tag{2.4}$$

ที่ $\{(a_{1j},a_{2j})\}_{j=1}^{25} = \begin{Bmatrix}(-32,-32),(-16,-32),(0,-32),(16,-32),(-32,-16),(-16,-16),(0,-16),(16,-16)\\(0,-16),(16,-16),(32,-16),....(-32,32),(-16,32),(-16,32),(0,32),(16,32)(32,32)\end{Bmatrix}$

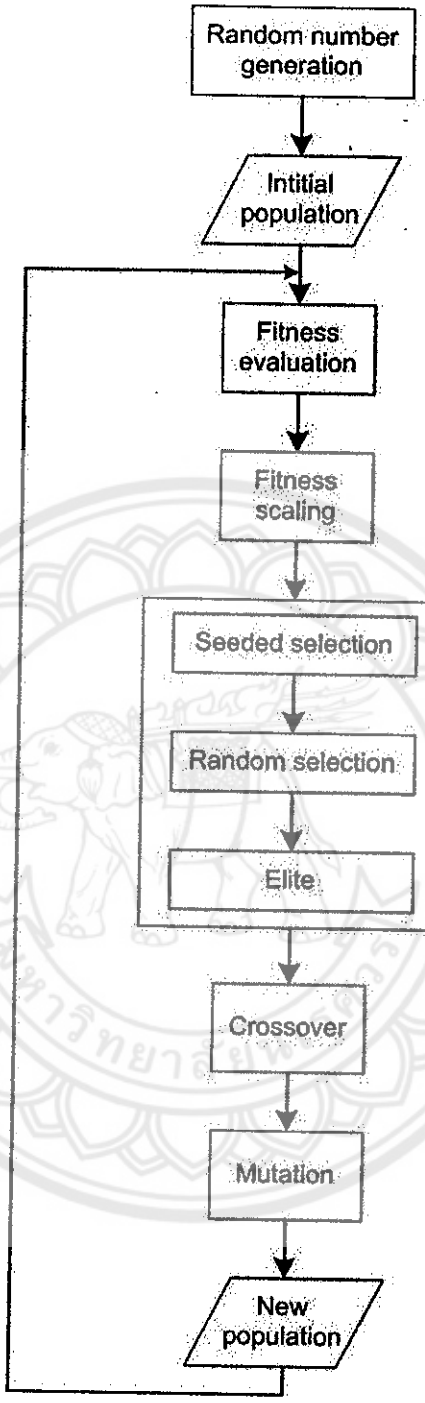
ตารางที่2.1 การหาพื้นที่ของคำตอบและการแก้ปัญหา

Function	Parameter Number	Solution		Parameter bound		Resoluton
		x_i	F_j	Lower	Upper	
F_1	3	0.00	0.00	-5.12	5.11	0.01
F_2	2	1.00	0.00	-2.05	2.047	0.001
F_3	5	-5.12	-30.00	-5.12	5.11	0.01
F_4	2	-32.00	1.00	-65.536	65.5350	0.001

(ที่มา : จากหนังสือ Intelligent Optimization Technech หน้า 28)

2.11 การใช้ Genetic Algorithm (GA)

Genetic Algorithm (GA) ที่นำมาประยุกต์ใช้นี้ถูกนิยามขึ้นโดย Grefenstelte [1986] กระบวนการแสดงดังภาพที่2.7



ภาพที่ที่ 2.7 แสดงการใช้กระบวนการทาง Genetic algorithm (GA)
(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 29)

แต่ละโครโมโซมในจำนวนประชากรของคำตอบจะถูกส่งเข้าสู่ขั้นตอนการประเมินค่าความแข็งแรงและสมบูรณ์ของโครโมโซม และหาโครโมโซมที่แข็งแรงและสมบูรณ์ที่สุดในการ

คำนวณหาคำตอบที่ดีที่สุดเท่าที่จะทำได้กำหนดให้ความสมบูรณ์และความแข็งแรงของโครโมโซมเท่ากับฟังก์ชัน

$$\text{fit} = (10000 - F) \tag{2.5}$$

กระบวนการการระบุขนาดความแข็งแรงนั้นจะต้องเป็นไปตามกฎของการประเมินค่าความแข็งแรงของโครโมโซม การให้ขนาดความแข็งแรงของโครโมโซมจะทำให้ได้ค่าของคำตอบที่ดีขึ้นในแต่ละรุ่นและจะนำไปสู่คำตอบที่ดีที่สุดเท่าที่จะหาได้กระบวนการในการคัดสรร(Selection) มีแตกต่างกันออกไป 3 กระบวนการ

1. การคัดสรรแบบสุ่ม(Random selection)
2. การคัดสรรแบบหว่านเมล็ดพันธุ์ (Seeded selection)
3. การคัดสรรแบบหัวกะทิ (Elite selection)

การคัดสรรแบบสุ่มนั้น จำนวนของโครโมโซมนั้นจะถูกสุ่มเลือกโดยผ่านกระบวนการในแต่ละรอบของรุ่นหนึ่งไปสู่รุ่นถัดไป เปอร์เซ็นต์ในการถูกเลือกนั้นจะถูกควบคุมด้วยพารามิเตอร์ที่เรียกว่า" Generation Gap" การคัดสรรแบบหว่านเมล็ดพันธุ์เป็นกระบวนการที่จำลองเทคนิค Roulette wheel เพื่อค้นหาโครโมโซมที่มีความบูรณ์แข็งแรงมากที่สุด การคัดสรรแบบหัวกะทินั้นเป็นกระบวนการที่จะทำให้เรามั่นใจมากขึ้นถึงจำนวนสมาชิกที่มีความสมบูรณ์ในแต่ละรุ่นที่ไม่เปลี่ยนแปลงในรุ่นถัดต่อๆไป

2.12 การใช้กระบวนการ Tabu search algorithm (TS)

ในที่นี้เราจะแทนที่คำตอบด้วยเวกเตอร์ เช่น $s = (x, \sigma)$ โดยจุดเริ่มต้นของเวกเตอร์ คือ $x(x_1 ..., x_2 ..., ... x_i)$ และจุดปลายของเวกเตอร์คือจุดในพื้นที่ที่หาคำตอบ จุดปลายของเวกเตอร์ $\sigma(\sigma_1 ..., \sigma_2 ..., \sigma_j)$ เป็นเวกเตอร์ที่แสดงถึงส่วนเบี่ยงเบนมาตราฐาน การหาคำตอบที่ใกล้เคียงจนได้คำตอบที่ดีที่สุดนั้นคือกระบวนการจากการแทนที่ค่า non-tabu x_i โดย

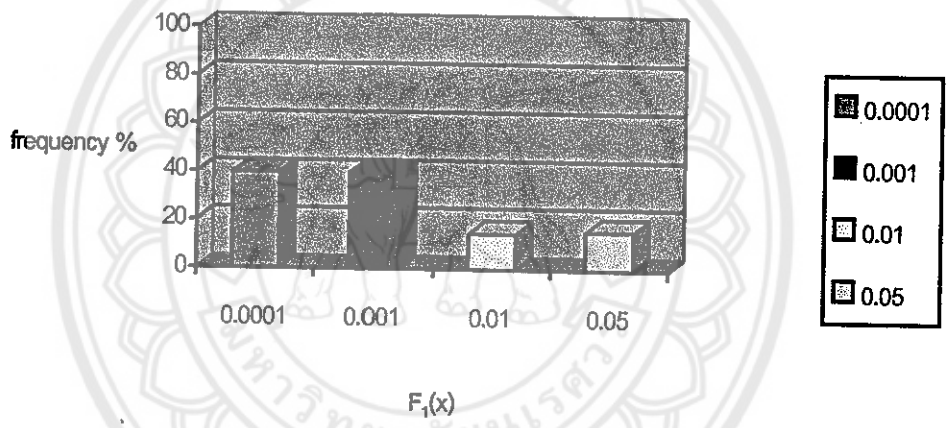
$$x_i = x + N(0, \sigma_i) \tag{2.6}$$

ที่ $N(0, \sigma_1)$ คือจำนวนการสุ่มแบบGaussianกับค่าเฉลี่ยของศูนย์และส่วนเบี่ยงเบนมาตราฐาน σ_i

ในการหาค่าของ σ_i ใช้กฎ 1/5 ของ Recenberg [Back *et al.*, 1991]

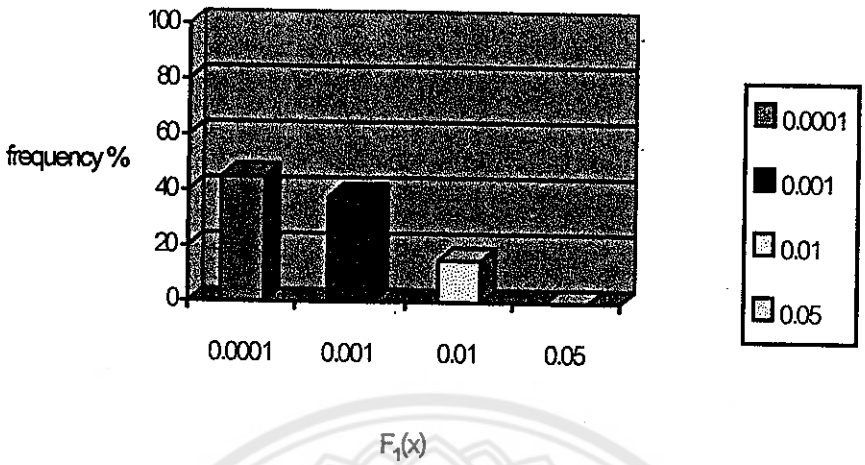
$$\sigma_i = \begin{cases} C_d \sigma_i, & \text{if } r(k) \quad 1/5 \\ C_i \sigma_j, & \text{if } r(k) \quad 1/5 \\ C_i & \text{if } r(k) = 1/5 \end{cases} \tag{2.7}$$

$r(k)$ คือ สัดส่วนของความสำเร็จในการเปลี่ยนค่า k ระหว่างค่า k ตัวสุดท้ายกับค่า k ที่เลือก และ C_i $1, C_d$ 1 เป็นตัวควบคุมอัตราการเพิ่มและอัตราการลดสำหรับค่าที่เปลี่ยนแปลงไป [Michalewicz , 1992] ในการทดสอบ Benchmarking สำหรับค่า C_i, C_d และ k ที่ใช้คือ $C_i = 1.2, C_d = 0.6, k = 10$ สำหรับในที่นี้ ถ้าจำนวนพารามิเตอร์ที่ใช้ในการหาคำตอบที่ดีที่สุดเท่าที่จะหาได้นั้นคือต้องมีพารามิเตอร์ที่น้อยกว่า 4 การบันทึกขอมูลความถี่ถูกใช้สำหรับการแยกประเภทของคำตอบว่าเป็นข้อห้ามหรือไม่ ในทางตรงกันข้าม ทั้งสิ่งที่เกิดขึ้นและความถี่ในการบันทึกข้อมูลนั้นถูกใช้ นี้คือเหตุผลเมื่อจำนวนพารามิเตอร์ที่น้อยกว่า 4 พารามิเตอร์ ทั้งหมดสามารถที่จะผ่านกระบวนการข้อห้าม(Tabu) ในจำนวนการทำซ้ำที่น้อย



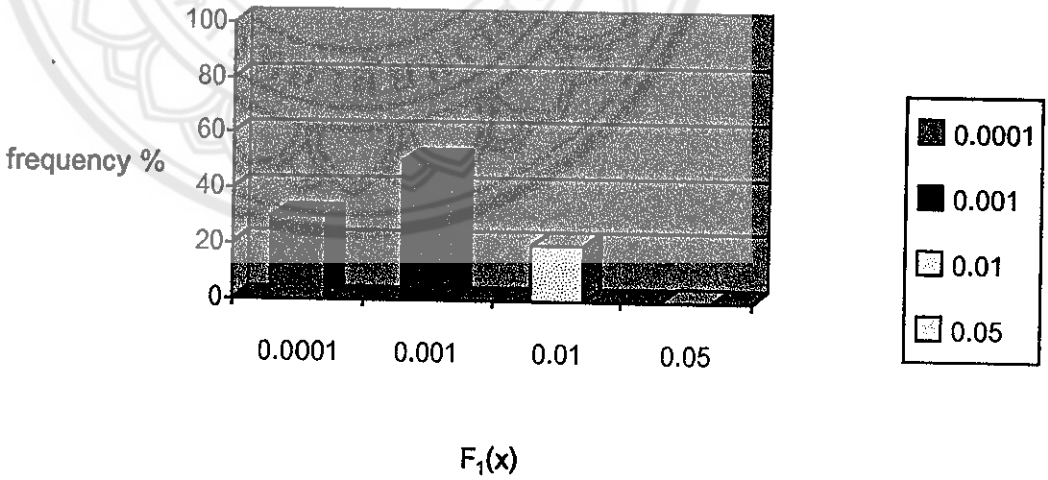
ภาพที่ 2.8 ฮิสโตแกรมของค่าฟังก์ชัน F_1 จากกระบวนการ F

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 34)



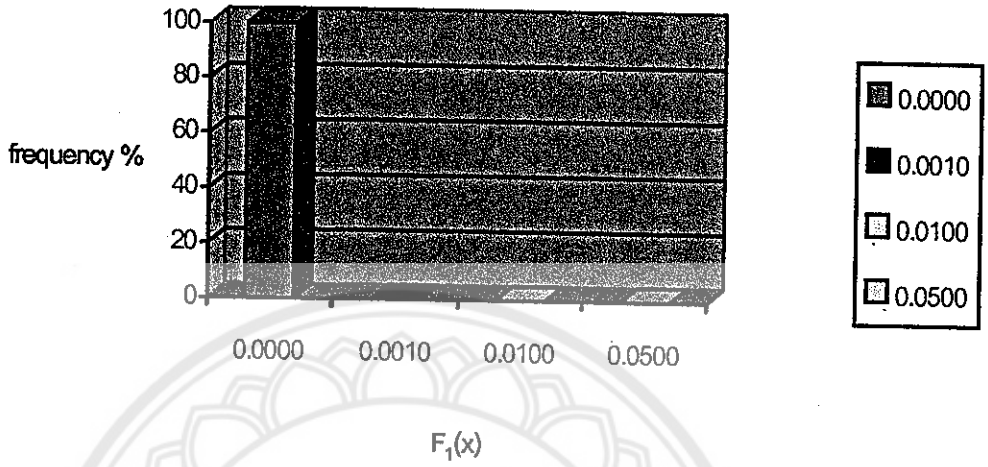
ภาพที่ 2.9 ฮิสโตแกรมของค่าฟังก์ชัน F_1 จากกระบวนการ TS

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal esource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 34)



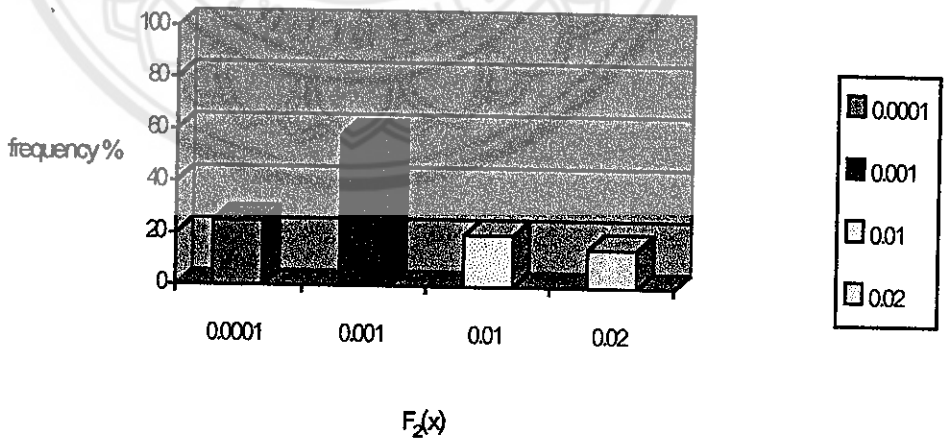
ภาพที่ 2.10 ฮิสโตแกรมของค่าฟังก์ชัน F_1 จากกระบวนการ SA

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 35)



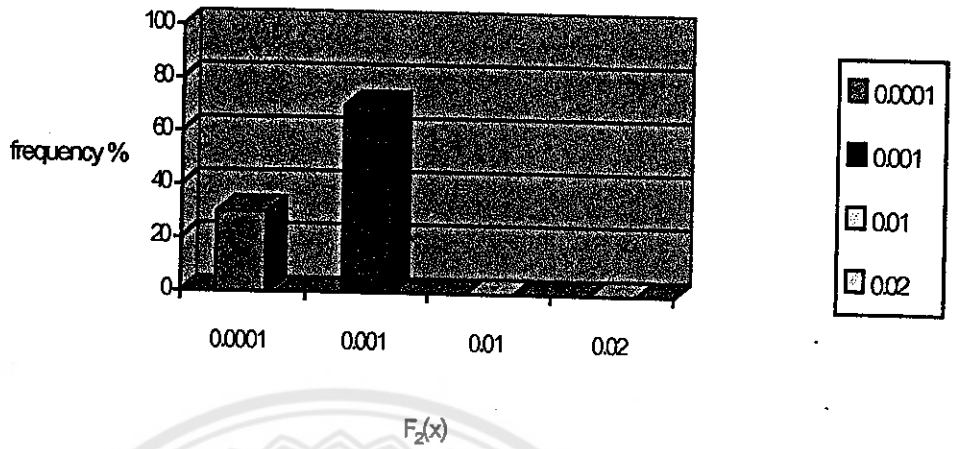
ภาพที่ 2.11 ฮิสโตแกรมของค่าฟังก์ชัน F_1 จากกระบวนการ NN

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 35)



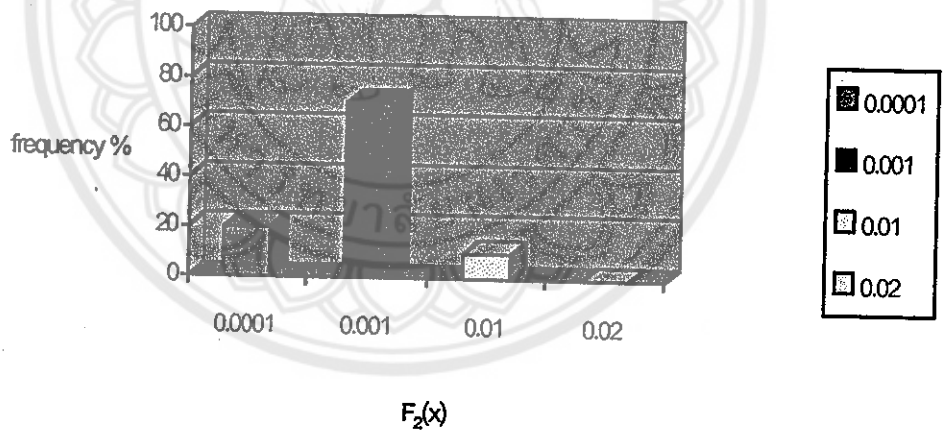
ภาพที่ 2.12 ฮิสโตแกรมของค่าฟังก์ชัน F_2 จากกระบวนการ GA

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 36)



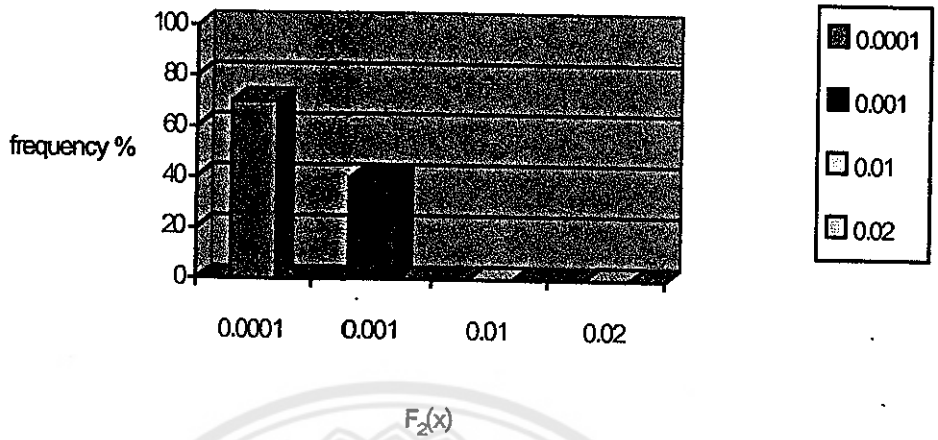
ภาพที่ 2.13 ฮิสโตแกรมของค่าฟังก์ชัน F2จากกระบวนการ TS

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 36)



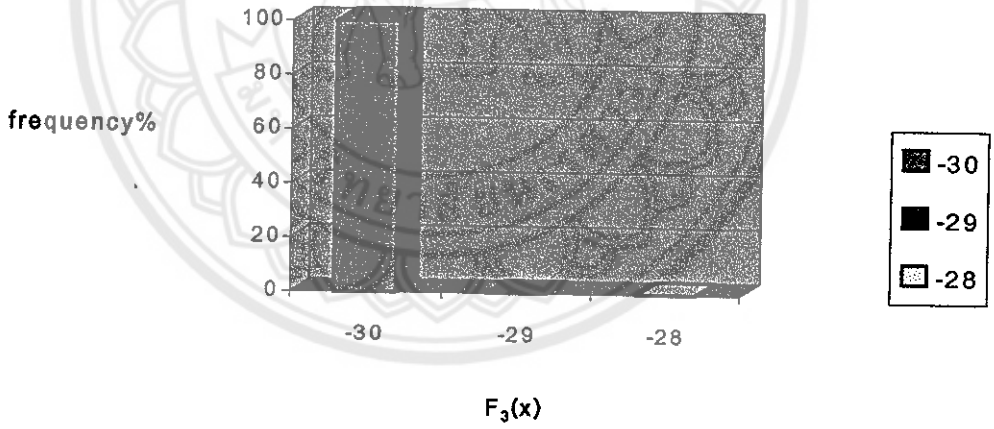
ภาพที่ 2.14 ฮิสโตแกรมของค่าฟังก์ชัน F_2จากกระบวนการ SA

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 37)



ภาพที่ 2.15 ฮิสโตแกรมของค่าฟังก์ชัน F_2จากกระบวนการ NN

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 37)

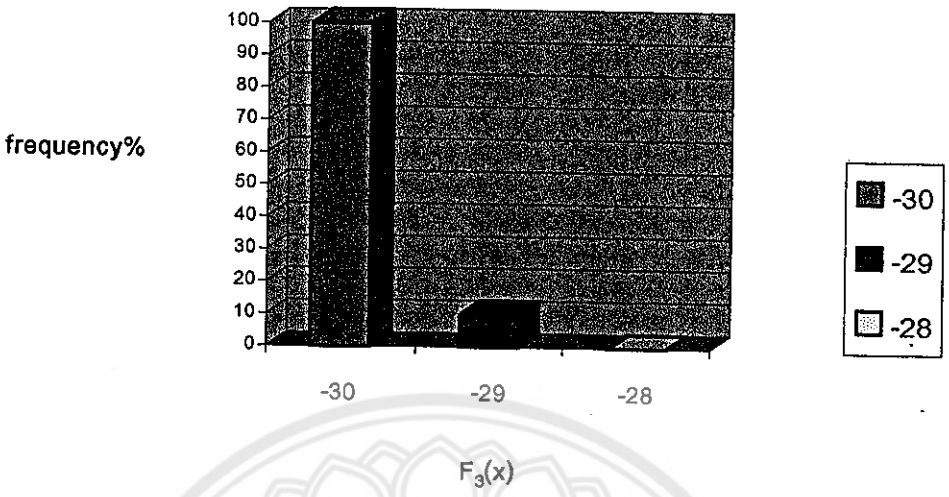


ภาพที่ 2.16 ฮิสโตแกรมของค่าฟังก์ชัน F_3จากกระบวนการ GA

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 38)

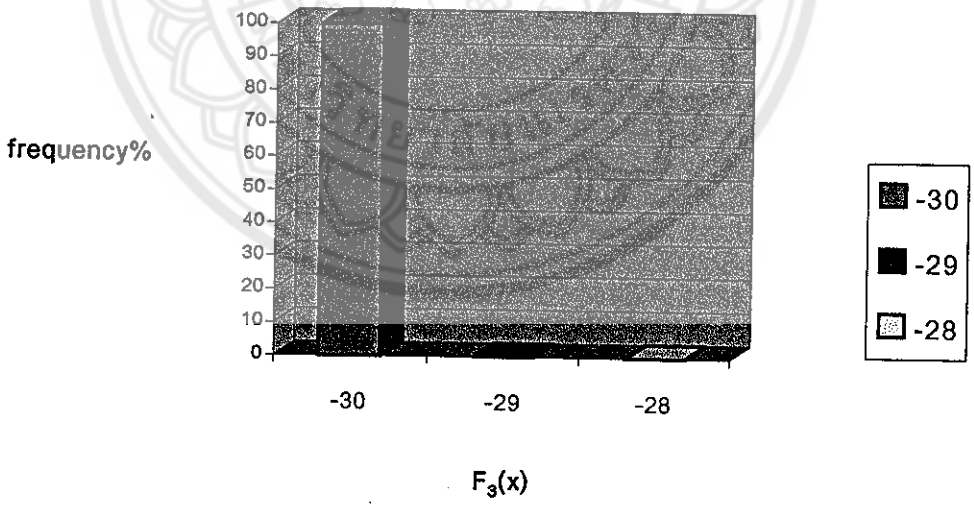
QA
76
.9
.B35
ป1719
2545

สำนักหอสมุด
22 ก.ค. 2547
4740403



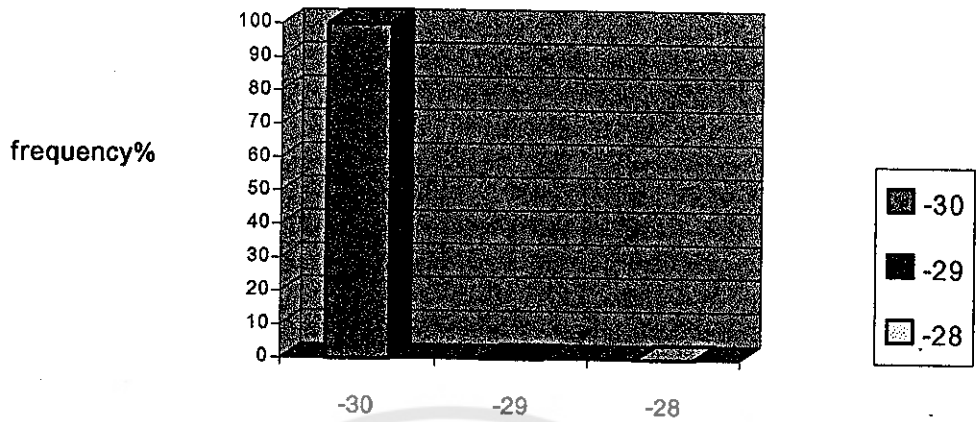
ภาพที่ 2.17 ฮิสโตแกรมของค่าฟังก์ชัน F_3จากกระบวนการ TS

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 38)



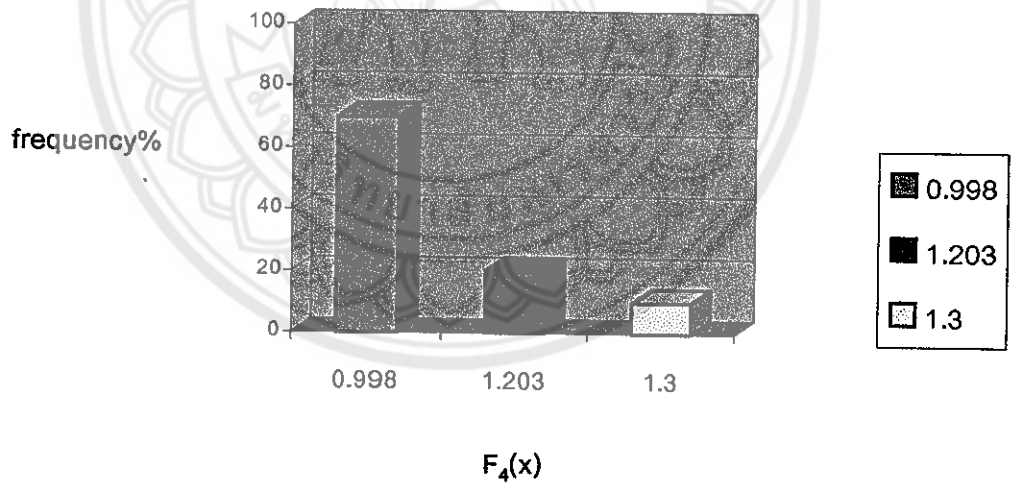
ภาพที่ 2.18 ฮิสโตแกรมของค่าฟังก์ชัน F_3จากกระบวนการ SA

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 39)



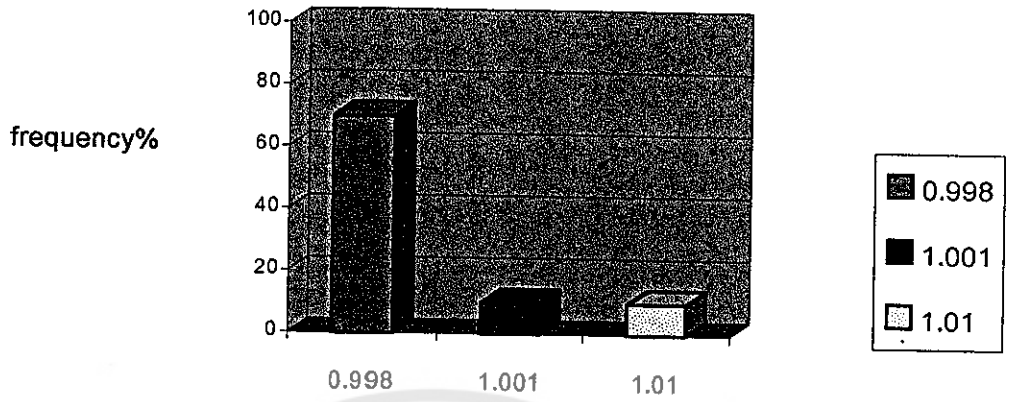
ภาพที่ 2.19 ฮิสโตแกรมของค่าฟังก์ชัน F_3 จากกระบวนการ NN

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 39)



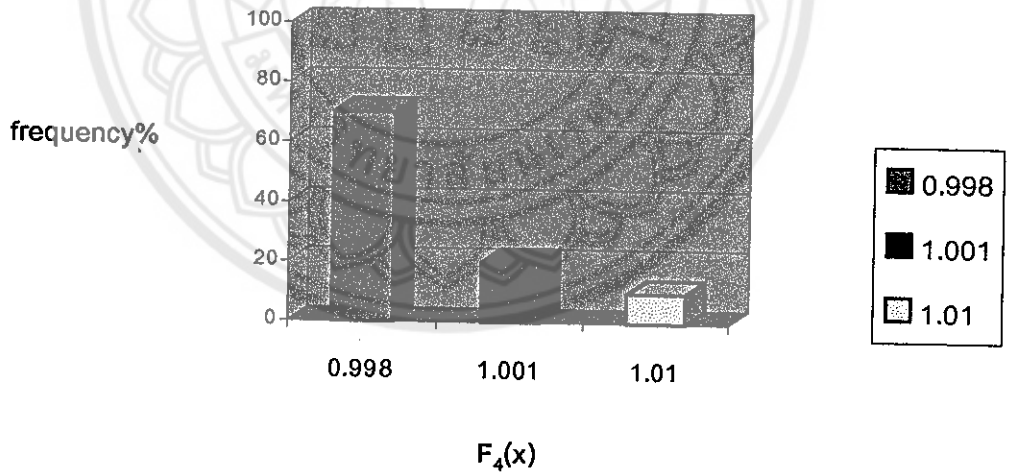
ภาพที่ 2.20 ฮิสโตแกรมของค่าฟังก์ชัน F_3 จากกระบวนการ GA

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 40)



ภาพที่ 2.21 ฮิสโตแกรมของค่าฟังก์ชัน F_4 จากกระบวนการ TS

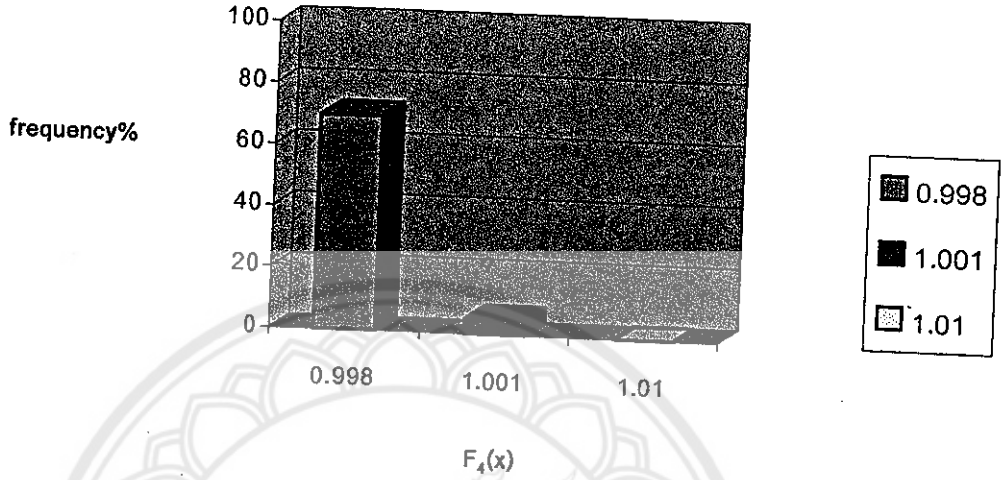
(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 40)



ภาพที่ 2.22 ฮิสโตแกรมของค่าฟังก์ชัน F_4 จากกระบวนการ SA

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 41)

2.23 F_4 NN



ภาพที่ 2.23 ฮิสโตแกรมของค่าฟังก์ชัน F_4 จากกระบวนการ NN (ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 41)

ตารางที่2.2 ค่าที่ได้จากกระบวนการหาคำตอบสำหรับฟังก์ชัน F_1 อย่างไม่ต่อเนื่อง

Evaluation Number	Genetic Algorithm	Tabu Search	Simulated Annealing	Neural Network
0	2.5390	14.3225	12.2531	52.3219
10	*	8.3762	3.3856	52.1961
20	*	1.4065	1.1289	51.2517
30	*	0.2324	0.4237	45.1706
40	*	0.0096	0.3296	23.6667
50	2.5390	0.0078	0.3454	4.8957
100	2.5390	0.0078	0.0467	0.0001
500	2.4762	0.00014	0.0015	0.0001
1000	0.7734	0.00014	0.0015	0.0001
2000	0.0021	0.00014	0.0015	0.0001
5000	0.0006	0.00014	0.0005	0.0001
10000	0.0006	0.00014	0.0005	0.0001

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 42)

ตารางที่2.3 ค่าที่ได้จากกระบวนการหาคำตอบสำหรับฟังก์ชัน F_2 อย่างไม่ต่อเนื่อง

Evaluation Number	Genetic Algorithm	Tabu Search	Simulated Annealing	Neural Network
0	3.5634	4.5067	3.6875	2480.68
10	*	0.3456	1.1214	1.1646
20	*	0.3456	1.0325	0.0297
30	*	0.3456	0.9889	0.0021
40	*	0.3456	0.9978	0.0006
50	3.5634	0.3456	0.6582	0.0004
100	1.5332	0.3456	0.5392	0.0003
500	0.1945	0.3078	0.1075	0.0003
1000	0.0031	0.0586	0.0877	0.0003
2000	0.0025	0.0017	0.0877	0.0003
5000	0.0025	0.0009	0.0019	0.0003
10000	0.0025	0.0005	0.0019	0.0003

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 42)

ตารางที่2.4 ค่าที่ได้จากกระบวนการหาคำตอบสำหรับฟังก์ชัน F_3 อย่างไม่ต่อเนื่อง

Evaluation Number	Genetic Algorithm	Tabu Search	Simulated Annealing	Neural Network
0	-16	-1	-10	-3
10	*	-6	-12	-7
20	*	-11	-13	-11
30	*	-19	-18	-16
40	*	-20	-19	-30
50	-16	-22	-26	-30
100	-19	-27	-28	-30
500	-27	-30	-30	-30
1000	-27	-30	-30	-30
2000	-28	-30	-30	-30
5000	-30	-30	-30	-30
10000	-30	-30	-30	-30

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 43)

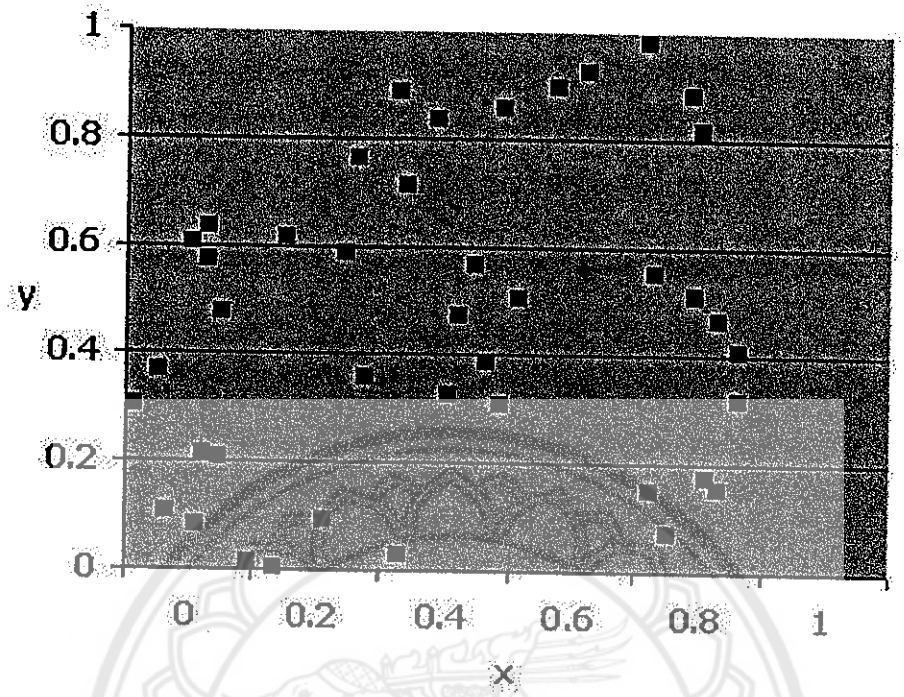
ตารางที่2.5 ค่าที่ได้จากกระบวนการหาคำตอบสำหรับฟังก์ชัน F_4 อย่างไม่ต่อเนื่อง

Evaluation Number	Genetic Algorithm	Tabu Search	Simulated Annealing	Neural Network
0	3.8756	499.1245	421.6523	499.9998
10	*	261.3594	125.7685	499.9998
20	*	56.3578	92.4536	499.9997
30	*	54.9811	68.2361	499.9997
40	*	51.2734	12.6724	499.9997
50	3.8756	19.5573	7.8743	499.9997
100	3.8756	10.1763	2.0643	499.9949
500	3.7965	1.1101	2.0643	499.9991
1000	3.7965	0.9980	0.9983	0.9991
2000	2.9922	0.9980	0.9983	0.9991
5000	0.9801	0.9980	0.9983	0.9991
10000	0.9980	0.9980	0.9981	0.9991

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 43)

2.13 ความแตกต่างของประสิทธิภาพในการหาคำตอบที่ดีที่สุดเท่าที่จะหาได้ในการแก้ปัญหาการเดินทางของเซลแมน

ปัญหาการเดินทางของเซลล์แมน Traveling Salesman Problem (TSP) นั้นเป็นปัญหาที่เป็นแบบอย่างของปัญหาเพื่อการหาคำตอบที่ดีที่สุดเท่าที่จะทำได้ (Optimization) สำหรับปัญหานี้การเดินทางของเซลล์แมนนั้นคือการเดินทางไปยังเมืองต่างๆที่กำหนดไว้ล่วงหน้า เมือง A,B,C,D วัตถุประสงค์ในการแก้ปัญหานี้คือหาระยะทางที่สั้นที่สุดในการเดินทางเพื่อไปในแต่ละเมืองต่างๆ แล้วกลับมายังตำแหน่งเดิม Genetic algorithm (GA) , Tabu Search (TS) , Simulated Annealing(SA) และ Neural network (NN) ถูกใช้ในการหาระยะทางในการเดินทางของเซลล์แมน เมืองทั้ง 50 เมืองนี้ได้มาจากการสุ่มในหน่วยสี่เหลี่ยม แผนที่แสดงดังภาพที่ 2.24



ภาพที่2.24 แผนที่ของเมืองทั้ง 50 เมือง

(ที่มา : SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141 หน้า 44)

การใช้กระบวนการ Genetic algorithm (GA) ในการแก้ปัญหากระบวนการ GA จะแสดงถึงการเดินทางทั้ง 50 เมืองด้วย) $P(C_1, C_2 L, C_{50})$ โดยที่ c คือ โครงสร้างแบบเส้น (binary string) C_i เป็นตำแหน่งของเมืองและเขียนตามอักขระของเมือง

ทุกๆเมืองสามารถถูกจัดเรียงได้ในทุกตำแหน่งของรายการ "แมนฮัตตัน" คือระยะทางระหว่างเมือง 2 เมืองที่อยู่ใกล้กัน กำหนดให้เป็น $d_{c_i c_{i+1}}$ และเรียกว่าเป็นระยะทางของเมืองทั้ง 2 เมือง การจัดเรียงตัวของทั้ง 50 เมือง ระยะทางทั้งหมดในรายการที่ต้องการจะเดินทางในแต่ละครั้ง,ความยาวสำหรับการเดินทางคือผลรวมของระยะทางของเมืองทั้งหมดที่อยู่ในรายการที่เราต้องการจะไปในแต่ละการจัดเรียงตัวนั้นเป็นลักษณะเฉพาะของแต่ละครั้ง ขอบเขตของการเดินทางคือผลรวมของระยะทางของเมืองทั้งหมดที่อยู่ในรายการที่เราต้องการจะไปในแต่ละการจัดเรียงตัวนั้นเป็นลักษณะเฉพาะของแต่ละครั้ง ขอบเขตของการเดินทางไปในแต่ละเมืองนั้น นั้นคือทำให้มั่นใจมากขึ้นเพราะทุกๆเมืองต้องปรากฏอยู่ในรายการในแต่ละครั้ง

กระบวนการได้เริ่มขึ้ ที่ 64 การจัดเรียงตัว ทั้งหมดเหล่านั้นถูกจัดเข้าในส่วนหนึ่งของจำนวนประชากรและความสมบูรณ์ สำหรับทุกๆการจัดเรียงตัวนั้นที่อยู่ในส่วนหนึ่งของจำนวนประชากร

จะถูกคำนวณขึ้น ในขั้นตอนต่อไป กระบวนการจะเลือก 8 การจัดเรียงตัวที่ดีที่สุดจากจำนวนประชากรมาเป็นครอบครัว (พ่อ,แม่) สำหรับประชากรที่อยู่ติดกัน อัตราส่วนสำหรับส่วนหนึ่งของประชากรคือ 0.25:0.20:0.15:0.10:0.10:0.10:0.05:0.07 ในส่วนของขั้นตอนสุดท้ายกระบวนการให้การจัดเรียงตัวในการจับคู่และใช้การดำเนินงานการตัดสลับและการกลายพันธุ์ขนาดของสายถูกเลือกโดยการสุ่มช่วงห่างของทั้ง 2 เมืองถูกแทนด้วย $[\frac{N}{2},\frac{N}{4}]$ และอัตราการกลาย พันธุ์ (Mutaton rate)คือ 0.01[Muhlenbein et al, 1988 Martina,1989]

การใช้กระบวนการ Tabu Search (TS)

กระบวนการ Tabu search นั้นจะแปลข้อความเป็นรหัส สำหรับ TSP กับการรวมกันของรายการเมืองที่ต้องการจะเดินทาง ค่าของรายการเมืองที่ต้องการจะไปคือ 1-50 ซึ่งเป็นรายการของเมืองทั้ง 50 เมืองในแผนที่การจัดเรียงต่อของตัวเลขดัชนีบนรายการที่ต้องการสำหรับการเดินทาง ระยะทางระหว่างเมืองทั้งสองเมืองนั้นถูกกำหนดให้เป็นแบบ "แมนฮัตตัน"ความยาวในการเดินทางคือผลกระทบของระยะทางทั้งหมดในรายการ ฟังก์ชันระยะทางในการเดินทางคือฟังก์ชัน COS ของ TSP ในแต่ละตัวเลขดัชนีต้องปรากฏอยู่ในรายการในแต่ละครั้ง

ในรายการข้อห้าม(Tabu)จะเป็นภาพที่ร่าง ก่อนที่จะมีการเปลี่ยนแปลงในรายการข้อห้ามจำเป็นที่ต้องมีการทดสอบเพื่อให้แน่ใจในคำสั่งในการเคลื่อนย้ายครั้งใหม่ จะไม่มีข้อห้ามของการแสดงคำตอบ การย้ายตำแหน่งใหม่ได้ถูกสร้างโดยการกำจัดตัวเลขดัชนีออกจากตำแหน่งในการรวมของรายการเมืองทั้งหมด และโดยการแทรกมันเข้าไปไว้ในระหว่างตัวเลขดัชนีอื่นๆ ถ้าการเดินทางอันใหม่ถูกสร้างโดยการแทรกเข้าไป จะต้องไม่มีในรายการข้อห้าม การเคลื่อนย้ายต้องได้รับการยอมรับ สิ่งที่เกิดขึ้นและความดีถูกใช้ในการหารายการข้อห้าม พารามิเตอร์สำหรับสิ่งที่เกิดขึ้นและความถี่คือ r = 0.5 และ f = 1.5 ตามลำดับ

2.14 การใช้กระบวนการ Simulated annealing (SA)

ปัญหาการเดินทางของเซลล์แมน (Traveling Salesman Problem),TSP ในกระบวนการ Simulated annealing algorithm (SA) มีการแก้ปัญหาที่เป็นไปในแนวทางเดียวกับกระบวนการ Tabu Search (TS)

ในกระบวนการ Simulated Annealing (SA) คือการนำตัวเลขดัชนี 1 ถึง 50 ในรายการซึ่งเป็นผลมาจากอุณหภูมิที่มีการเปลี่ยนแปลงอยู่ตลอดเวลา อุณหภูมิที่มีการเปลี่ยนแปลงอยู่ตลอดนี้ได้จาก

ตารางการอบอ่อน (annealing) ตารางการอบอ่อนและพารามิเตอร์เป็นไปตามดังนี้[Kirkpatrick 1983]

1. อุณหภูมิเริ่มต้น $T_0 = 10$ และความยาวของช่วงเวลา $L_0 = 50$

2. ความร้อนที่เพิ่มขึ้นโดยการเปลี่ยนแปลงอุณหภูมิถึง $T = T_0/0.8$ และให้ ค่า $L = 50$ จนกระทั่งความแปรเปลี่ยนของฟังก์ชันต้นทุน (function cost) $\langle 0.05$

3. ความเย็นที่ลดลงโดยการเปลี่ยนแปลงอุณหภูมิลง $T = 0.95T$ และให้ค่า $L = 50$ เมื่อเปอร์เซ็นต์ของการจัดเรียงตัวนั้นที่ยอมรับได้ต้องมากกว่า 50 %

4. ความเย็นที่ลดลงอย่างช้าๆ โดยการเปลี่ยนแปลงอุณหภูมิถึง $T = 0.95T$ และเปลี่ยนให้ L ถึง 16L เมื่อถึงจุดที่มีการจัดเรียงตัวที่ยอมรับได้

2.16 การใช้กระบวนการ Neural Network

ปัญหาในการเดินทาง 50 เมืองของเซลล์แมนเป็นจำนวนของอาเรย์ขนาด 50×50 ค่าดัชนีของแถวของอาเรย์แสดงถึงเมืองต่างๆ ดัชนีของคอลัมน์แสดงถึงตำแหน่งที่ต้องการจะเดินทางของเซลล์แมนในรายการ ข้อมูลของอาเรย์สามารถใส่ค่าได้ที่ 1 กับ 0 ข้อจำกัดในการเดินทางไปในแต่ละเมืองนั้นจะมีเพียง 1 ค่าในทุกๆแถวและทุกๆหลัก สำหรับ 50 เมือง ผลรวมทั้งหมดทั้ง 50 กลุ่มของระบบประสาทจำเป็นที่จะต้องแทนที่ด้วยขนาดของอาเรย์ที่ 50×50 ในแต่ละระบบเส้นประสาทคือการใส่ค่าลงไปให้กับตัวแปรอาเรย์

Hopfield network คือ สำหรับการแก้ปัญหาขนาดเล็กที่เหมาะสมสำหรับ TSPs ประกอบไปด้วย 10 เมืองหรือมากกว่านั้นเล็กน้อย [Hopfield and Tank, 1985] สำหรับปัญหา ที่มากกว่า 10 เมืองจะเป็นเหตุให้ยุ่งยาก Potts neural network [Catster and Bo, 1989; 1990] จะแสดงถึงกระบวนการการหาคำตอบที่ดีและ Potts neural network สามารถหาคำตอบได้ด้วยฟังก์ชัน

$$V_{ia} = \frac{e^{u_{ia}}}{\sum_{b} e^{u_{ib}}} \qquad (2.8)$$

ที่ u_{ia} เป็น internal potential ของ ระบบประสาทของลำดับที่ i^{th} กำหนดให้

$$u_{ia} = -\frac{\partial E}{\partial v_{ia}} \qquad (2.9)$$

ผลการทดลองการใช้กระบวนการต่างๆในการแก้ปัญหา Traveling Salesman Problem (TSP)

ประสิทธิภาพทั้งหมดของกระบวนการแก้ปัญหาเพื่อหาคำตอบที่ดีที่สุดที่จะหาได้ แสดงดังตารางที่ 2.6

ตารางที่ 2.6 ระยะทางในการเดินทางที่ได้รับ

Genetic algorithm	Tabu search	Simulated annealing	Neural network
5.58	5.68	6.80	6.61

สรุป

ในส่วนของบทนี้ได้ใช้ทฤษฎีของทั้ง 4 เทคนิคในการหาคำตอบที่ดีสุดเท่าที่จะหาได้ คือ Genetic algorithm, Simulated Annealing , Tabu search, Neural network ในการทดสอบเพื่อหาคำตอบที่ดีที่สุดของจำนวนตัวเลขของฟังก์ชันที่นำมาทดสอบได้พิสูจน์และแสดงให้เห็นถึงประสิทธิภาพของแต่ละเทคนิค ในการแก้ปัญหาการเดินทางของเซลแมนทั้ง 50 เมืองนั้นได้ถูกนำมาปืนปัญหาเพื่อใช้ในการทดสอบถึงประสิทธิภาพของแต่ละวิธีการในการหาคำตอบที่ดีที่สุดเท่าที่จะหาได้

2.17 บทความเกี่ยวกับการประยุกต์ใช้ Genetic algorithm การแก้ปัญหาต่างๆ

1. Runwei Chenga*, Mitsuo Genb, Yasuhiro Tsujimurab ,A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies, ,Computers & Industrial Engineering 36 (1999) 343±364
2. SOU-SEN LEU* and TZUNG-HENG HUNG,A genetic algorithm-based optimal resource-constrained scheduling simulation model,Construction Management and Economics(2002) 20, 131-141
3. Runwei Cheng[a,*], Mitsuo Gen[b], Yasuhiro Tsujimura[b], A tutorial survey of job-shop scheduling problems using genetic algorithms,part II: hybrid genetic search strategies, Computers & Industrial Engineering 36 (1999) 343-364
4. X. Cai[a,*], K.N. Li[b], A genetic algorithm for scheduling staff of mixed skills under multi-criteria, European Journal of Operational Reseach 125 (2000) 359-369

5. DONG HYUN BEAK and WAN CHUL YOON, Co-evolutionary genetic algorithm for multi-machine scheduling:coping with high performance variability, Int.J. PROD.RES.,2002, VOL.40,NO.1, 239-254

6. P.Pongcharoen[a],C.Hicks[a,*],P.M.Braiden[a],D.J. Stewardson[b], Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products, Int.J.Production Economics 78 (2002) 311-322

7. P.Pongcharoen[a],C.Hicks[a,*],P.M.Braiden[a],D.J. Stewardson[b], Applying designed experiments to optimize the performance of genetic algorithm used for scheduling complex products in the capital goods industry, Journal of Applied Statistics,Vol.28,No.34,2001,441-455

8. G C Onwubolu, Manufacturing cell scheduling using genetic algorithms, PO Box AC939,Ascot,Bulawayo,Zimbabwe

9. SOU-SEN LEU and CHUNG-HUEI YANG, A genetic-algorithm-based resource-constrained construction scheduling system, Construction Management and Economics(1999) 17,767-776

10. Liu Min*,Wu Cheng, A genetic algorithm for minimizing the makespan in the case of scheduling identical paralle machines, Artificial Intelligence in Engineering 13 (1999) 399-403

บทที่ 3
วิธีการดำเนินงาน

3.1 ศึกษาหลักการและทฤษฎีของ Genetic Algorithm(GA) (P.Pongcharoen 2001)

เพื่อทำความเข้าใจในกระบวนการของGAทั้งหมดทำให้ทราบว่ามีฟังก์ชันต่างๆดังนี้

3.1.1 ฟังก์ชั่นการเลือกสุ่มทำหน้าที่เป็นผู้คัดเลือกในทุกๆส่วนของโปรแกรมที่มีการเลือกสุ่ม ซึ่งการเลือกสุ่มแต่ละครั้งนั้นจะมี Seed เป็นตัวกำหนดตารางสุ่มของโปรแกรม

3.1.2 ฟังก์ชันการสร้างต้นแบบพันธุกรรม(พ่อแม่) ทำหน้าที่สุ่มสร้างแต่ละโครโมโซมโดยทำการสร้างที่ละ1ยีนซึ่งแต่ละยีนจะต้องไม่มียีนใดที่ซ้ำกันเลย แต่ทว่าในแต่ละโครโมโซมสามารถซ้ำกันได้เนื่องจากเป็นการเลียนแบบธรรมชาติซึ่งอาจจะมีฝาแฝดได้จึงทำให้เกิดการซ้ำกันในโครโมโซมได้

3.1.3 ฟังก์ชัน Genetic Operation ทำหน้าที่เป็นการเลียนแบบกระบวนการสร้างประชากรใหม่ในแต่ละรุ่น ซึ่งจะมีการแบ่งย่อยออกเป็น 2 กระบวนการคือCrossover(การตัดสลับ)และ Mutation(การ-กลายพันธุ์)

3.1.4 ฟังก์ชันการคำนวณค่าผลลัพธ์ของโครโมโซมเพื่อนำไปวัดค่าความแข็งแรงของแต่ละโครโมโซม

3.1.5 ฟังก์ชัน Roulette Wheel ทำหน้าที่คัดเลือกโครโมโซมโดยมีความแข็งแรงของโครโมโซมเป็นปัจจัยในการกำหนดความน่าจะเป็นที่จะได้รับการัดเลือกของแต่ละโครโมโซม

3.1.6 ฟังก์ชัน Chromosome Selection(การสรรหาโครโมโซม) ทำหน้าที่สุ่มเลือกโครโมโซมเข้าในกระบวนการ Genetic Operation

3.2 ศึกษาปัจจัยต่างๆ ที่เกี่ยวข้องกับกระบวนการ (P.Pongcharoen 2001)

จากการในส่วนของทฤษฎี Genetic Algorithm พบว่ามีปัจจัยต่างๆที่ส่งผลต่อผลของการคำนวณดังนี้

Population size (จำนวนประชากร) ตัวแปรนี้จะเป็นตัวกำหนดจำนวนต้นแบบที่สร้างขึ้นขอบเขตการสุ่มเลือกประชากร จำนวนในการคัดเลือกในแต่ละรุ่น

3.2.1 Number of Gene (จำนวนของยีนในแต่ละโครโมโซม) ตัวแปรนี้จะเป็นตัวกำหนดความยาวของยีนซึ่งต้องสัมพันธ์กับฟังก์ชันที่ทำการ Code มาสู่ Genetic form

3.2.2 Number of Generation (จำนวนช่วงอายุ) จะเป็นตัวกำหนดว่าโปรแกรมจะต้องวนทำงานซ้ำทั้งหมดกี่รอบ

3.2.3 Crossover rate (อัตราการตัดสลับ) ตัวแปรนี้จะกำหนดให้รู้ว่าในแต่ละครั้งของการให้กำเนิดใหม่นั้นจะต้องทำการตัดสลับทั้งหมดกี่โครโมโซมจากจำนวนประชากรทั้งหมด

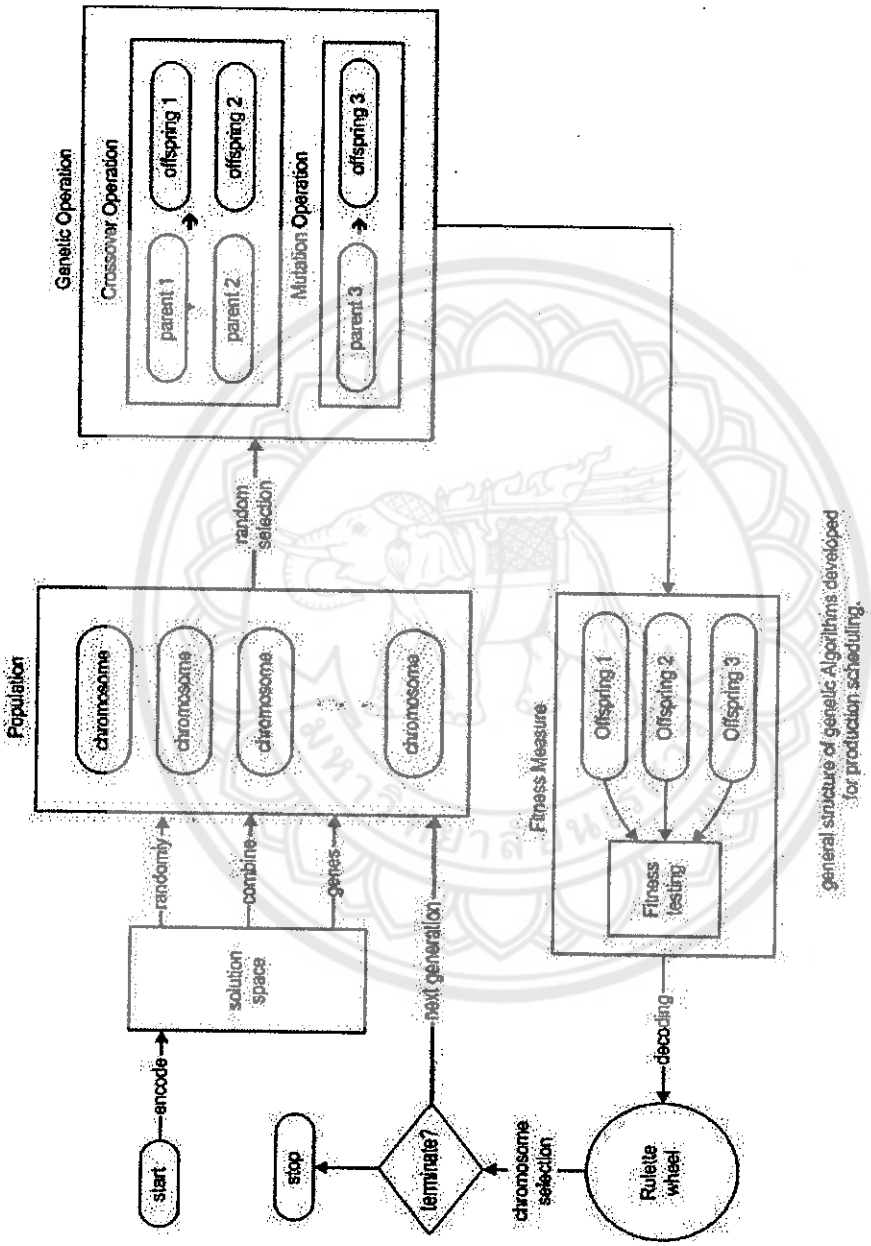
3.2.4 Mutation rate (อัตราการกลายพันธุ์) ตัวแปรนี้จะกำหนดให้รู้ว่าในแต่ละครั้งของการให้กำเนิดใหม่นั้นจะต้องทำการกลายพันธุ์ทั้งหมดกี่โครโมโซมจากจำนวนประชากรทั้งหมด

3.2.5 Selection Based on (ฐานการคัดเลือก) ตัวแปรนี้จะกำหนดวิธีการคัดเลือกของ Roulette Wheel ว่าจะเลือกจากลูกเพียงอย่างเดียว(Regular Sampling Space) หรือคัดเลือกจากพ่อแม่รวมกับลูก (Enlarged sampling space)

3.2.6 Random seed (ตัวกำหนดชุดเลขสุ่ม) ตัวแปรนี้หากไม่มีการกำหนดแล้วเราจะไม่สามารถคำนวณให้ได้ผลลัพธ์เดิมได้เลยเพราะ Seed จะเป็นค่าที่กำหนดชุดเลขสุ่มออกมาหากกำหนด Seedเป็นค่าเดิมจะทำให้เลขสุ่มที่ออกมาเป็นค่าเดิมด้วย

3.2.7 Genetic operation method (รูปแบบของกระบวนการทางพันธุกรรม) ซึ่งตัวแปรชนิดนี้จะเป็นการเลือกวิธีการตัดสลับ (Crossover) และวิธีการกลายพันธุ์ (Mutation) ในรูปแบบต่างๆซึ่งได้อ้างอิงจากผลงานวิจัยที่เกี่ยวกับการพัฒนา Genetic Algorithm

3.3 แผนภาพการทำงานของ GA



* P.Pongcharoen et al./ Int. J. Production Economics 78 (2002) 311-322
general structure of genetic Algorithms developed for production scheduling.

ภาพที่ 3.1 แผนภาพการทำงานของ GA

3.4 ขั้นตอนการทำงานของ Genetic Algorithm

3.4.1 การแปลงรูปแบบ(Encode)ของปัญหา ที่จะนำมาเข้าทำการวิเคราะห์ด้วยกระบวน Genetic algorithm โดยจัดเป็นรหัส ในขั้นต้นก่อนที่จะถูกจัดเข้าระบบการทำงาน

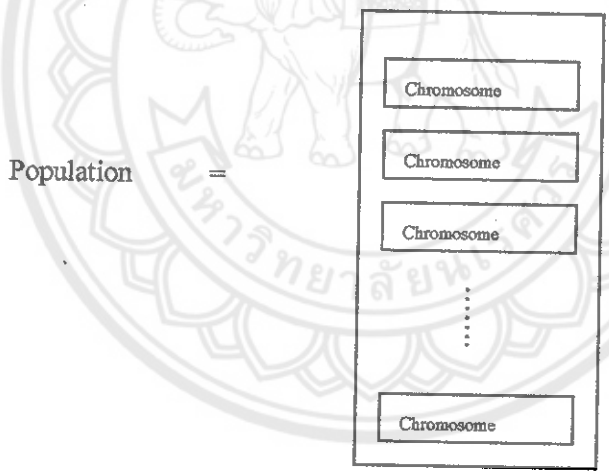
3.4.2 นำ รหัสที่ได้(Encode) มาเป็น Gene โดยที่เป็นส่วนประกอบของ Chromosome อีกที่หนึ่ง

3.4.3 ทำการสร้าง Chromosome โดยมีการสุ่มเลือก Gene ที่ถูกแปลกจากตัวปัญหามาเป็นรหัส(Encode) ซึ่งในส่วนของ Chromosome จะถูกประกอบไปด้วย Gene หลายๆตัว และไม่จำกัดว่าจะต้องเรียงลำดับของ Gene

เช่น

Chromosome = Gene2+Gene1 +Gene3 ...+GeneN

3.4.4 Population คือ ส่วนของประชากรที่บรรจุอยู่ภายในระบบของ Genetic โดยที่ประชากรในแต่ละตัวจะถูกเรียกว่า Chromosome



ภาพที่ 3.2

3.4.5 ทำการสุ่มเลือก Chromosome ที่อยู่ภายใน Population

3.4.6 Genetic Operation จะถูกประกอบไปด้วย 2 ส่วน คือ Crossover Operation และ Mutation Operation

3.4.6.1 ส่วนที่1: Crossover Operation จะมีชนิดของการทำงานอยู่ด้วยกัน 3 แบบ

ก. One-point crossover (Murata & Ishibuchi,1994)

ข. Position based crossover (Syswerda,1991)

ค. Two-point center crossover (Murata & Ishibuchi,1994)

3.4.6.2 ส่วนที่2: Mutation Operation จะมีชนิดของการทำงานอยู่ด้วยกัน 7 แบบ

ก. Centre inverse mutation (Tralle,2000)

ข. Inversion mutation (Goldberg,1989)

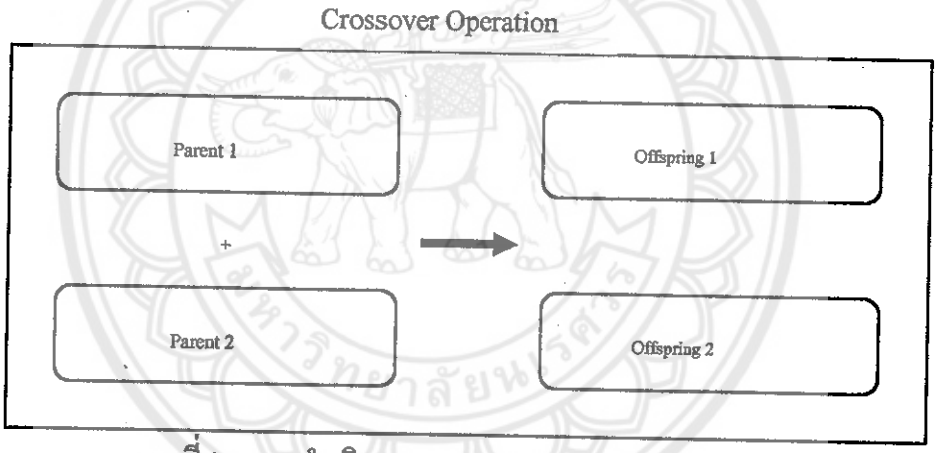
ค. Two-operation adjacent swap (Murata & Ishibuchi,1994)

ง. Three-operation adjacent swap (Murata & ishibuchi,1994)

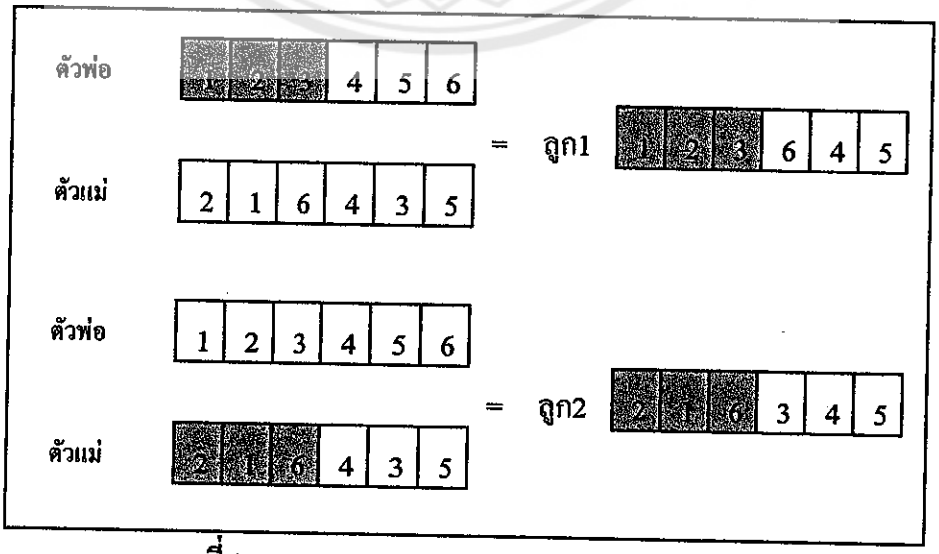
จ. Two-operation random swap (Murata & Ishibuchi,1994)

ฉ. Three-operation random swap (Murata & Ishibuchi,1994)

ช. Shift operation mutation (Murata & Ishibuchi,1994)



ภาพที่ 3.3 การกำเนิดลูกจาก กระบวนการ Crossover Operation

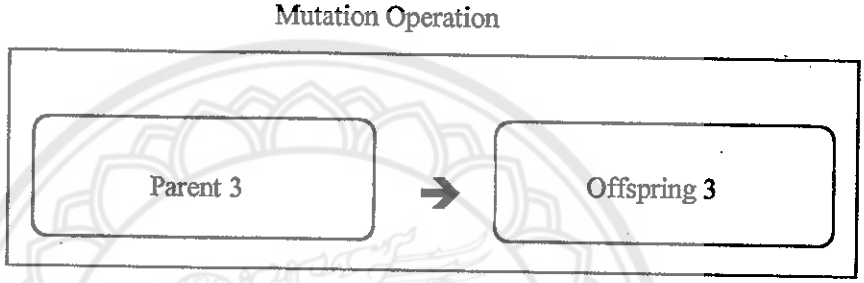


ภาพที่ 3.4 One-point crossover (Murata & Ishibuchi,1994)

แสดงถึง การผสมพันธ์ ระหว่าง

ตัวพ่อ(Parent1) + ตัวแม่(Parent2) = ลูก 2 ตัว

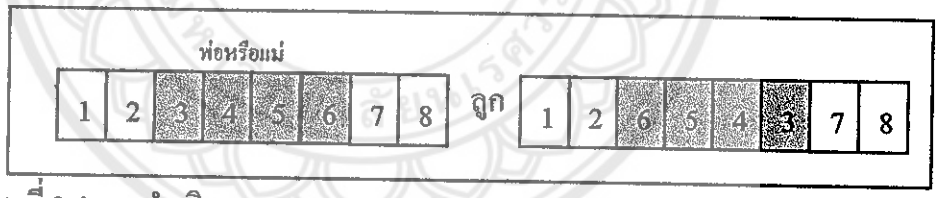
ในส่วนที่เป็นลูก แต่ละตัวจะมีส่วนประกอบที่เป็นของ ตัวพ่อหรือตัวแม่ ติดอยู่ที่ตัวลูกเสมอและ Gene ที่ประกอบอยู่ภายใน Chromosome ยังมีอยู่ครบทุกตัว เว้นแต่มีการสลับตำแหน่งที่อยู่ของ Gene โดยใช้กลวิธีที่มีหลักการของ Crossover Operationในการสลับตำแหน่งของ Gene ที่ประกอบอยู่ภายใน ตัวของ Chromosome ตามวิธีทั้ง 3 แบบ ของกระบวนการ Crossover



ภาพที่ 3.5 การกำเนิดลูกจาก กระบวนการ Mutation Operation

แสดงถึง การกลายพันธ์

ตัวพ่อหรือตัวแม่ (Parent) = ลูก 1 ตัว

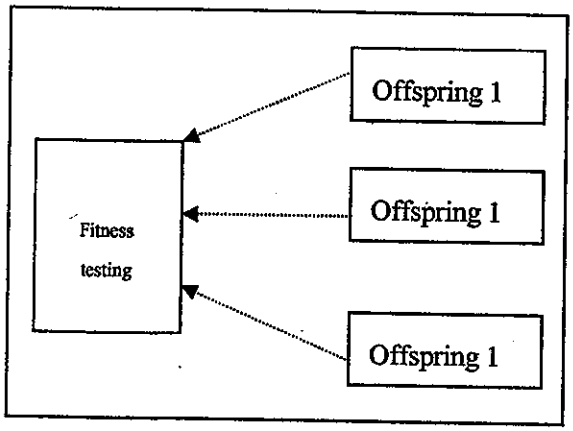


ภาพที่ 3.6 การกำเนิดลูกจาก กระบวนการ Inversion mutation (Goldberg,1989)

ส่วนประกอบภายในที่เป็น Gene ยังมีครบอยู่ทุกตัว เว้นแต่มีการสลับตำแหน่งที่อยู่ของ Gene โดยใช้กลวิธีที่มีหลักการของ Mutation Operation ในการสลับตำแหน่งของ Gene ที่ประกอบอยู่ภายใน ตัวของ Chromosome ตามวิธีทั้ง 7 แบบ ของกระบวนการ Mutation

3.5 Fitness Measure

Fitness Measure คือ ขั้นตอนในการคำนวณหาค่าตามที่ได้กำหนดในขั้นต้นว่าต้องการค่าที่มากที่สุด หรือค่าที่น้อยที่สุด โดยจะมีการให้เปอร์เซ็นต์ความแข็งแรงกับค่าที่คำนวณได้ ถ้าระบบ

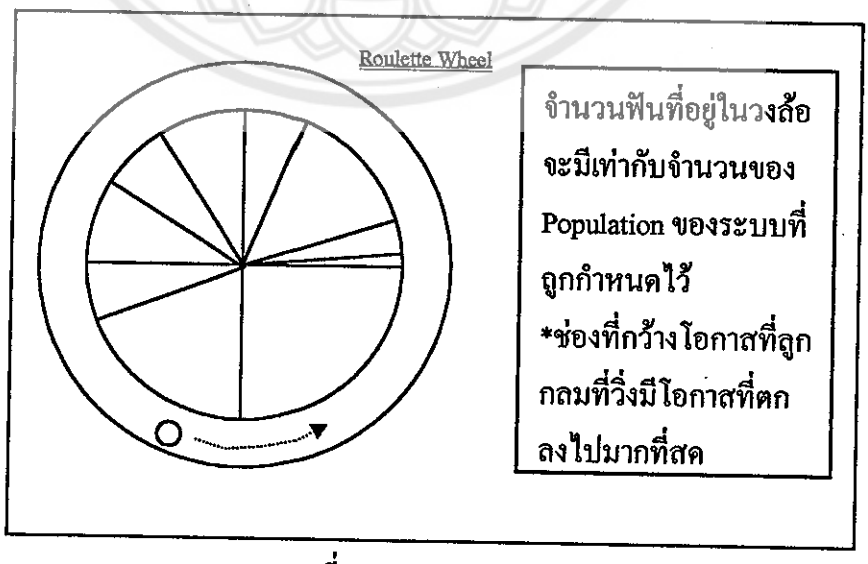


ภาพที่ 3.7 Fitness measure

การคำนวณถูกกำหนดให้หาค่าที่น้อยที่สุดค่าของลูก(Offspring)ตัวไหนที่ได้ถูกคำนวณออกมาแล้วได้ค่าที่น้อยสุดก็จะมีความแข็งแรงที่สุดพร้อมกับมีโอกาสที่จะกลายไปพ่อพันธ์ ที่ดีในรอบการคำนวณต่อไป ฉะนั้น Fitness Measure คือ การวัดหาลูกที่มีประสิทธิภาพในการเข้าผสมพันธ์และการกลายพันธ์เพื่อให้ได้ค่าที่เหมาะสมที่สุด

3.6 Roulette Wheel

Roulette Wheel คือ ล้อที่มีฟันสำหรับให้ลูกที่หมุนไปตกโดยในแต่ละช่องจะมีตัวเลขอยู่ภายในและระยะของฟันที่มีความกว้างไม่เท่ากันนี้ จะทำให้การหมุนในแต่ละรอบ มีโอกาสที่จะตกไปในแต่ละช่องไม่เท่ากัน แต่โอกาสที่จะตกเข้าไปที่ระยะของฟันที่ห่างมากที่สุดในล้อก็มีมากที่สุดเช่นเดียวกัน

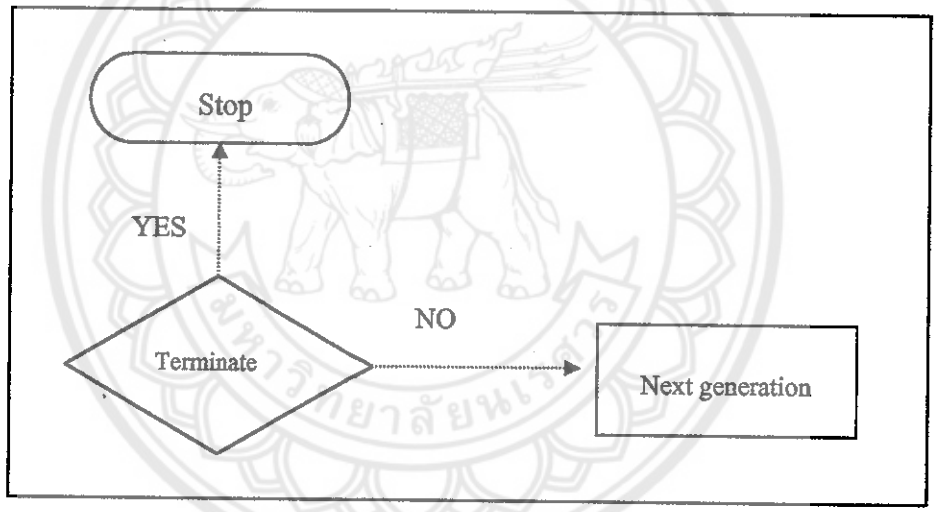


ภาพที่ 3.8 Roulette Wheel

จากรูปข้างต้นจะเห็นได้ว่า ค่าที่คำนวณได้ของลูก(Offspring)จะมีความแข็งแรงที่ไม่เท่ากัน ใน Function การทำงานของ Roulette wheel จะเป็นการสุ่มเลือกลูกแต่ละตัวโดยให้โอกาสในการสุ่มออกมาที่ไม่เท่ากัน จะมีการให้โอกาสกับลูก(Offspring)ที่มีคุณสมบัติที่สอดคล้องกับการกำหนดระบบการคำนวณ ว่าต้องการค่าที่มากสุดหรือน้อยที่สุด เมื่อทำการคัดเลือก Chromosomeลูกที่เกิดขึ้น จากการ Crossover Operation และการ Mutation Operation แล้วจะมีการส่ง Chromosome ลูกที่เกิดขึ้นไปเป็นพ่อพันธ์และแม่พันธ์ที่ดีต่อๆไป

3.7 Decoding

Decoding คือ การแปลง Encode กลับเพื่อทำการคำนวณหาค่าที่ได้จาก Gene ภายในตัว Chromosome



ภาพที่3.9 Decoding

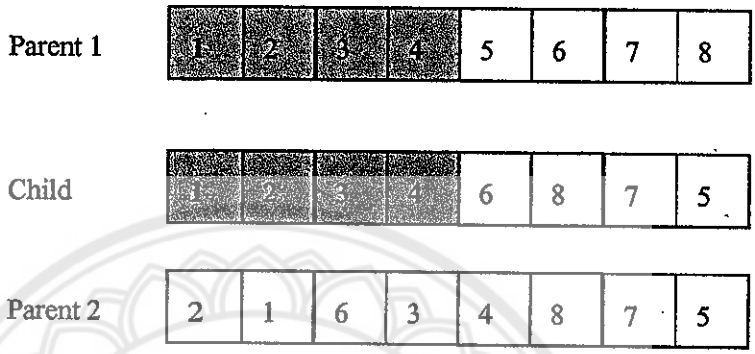
3.8 การทำงานในขั้นที่ต้องมีการตัดสินใจ

คือ มีการตรวจสอบว่าได้มีการจำกัดรอบของการคำนวณไว้ที่กี่รอบถ้าครบก็ทำการหยุดการทำงาน ถ้ายังไม่ครบก็เข้ารอบการทำต่อไปจนกว่าจะครบ

3.9 กระบวนการทำงานที่ใช้เป็น FUNCTION ภายในโปรแกรม

3.9.1 กระบวนการของ Crossover มีทั้งหมด 3 วิธี

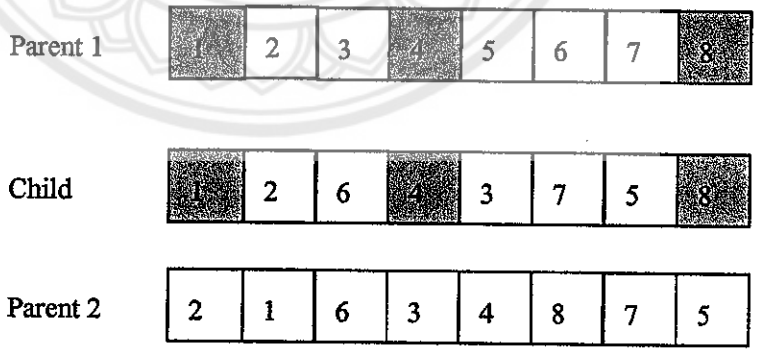
3.9.1.1 One-point crossover (Murata & Ishibuchi,1994)



ภาพที่ 3.10 One-point crossover (Murata & Ishibuchi,1994)

หนึ่งจุดถูกเลือกอย่างสุ่มๆ เพื่อแบ่งพ่อหรือแม่ตัวใดตัวหนึ่งออกเป็น 2 ส่วน ส่วนแรกจะถูกตัดออกมาเป็นลูกโดยตรง gene ที่เหลือของลูกจะถูกเติมเต็มตามลำดับจากอีกพ่อหรือแม่หนึ่ง กระบวนการนี้ถูกกระทำอีกครั้งหนึ่งแต่เป็นทางกลับกัน เพื่อให้ได้ลูกตัวที่ 2

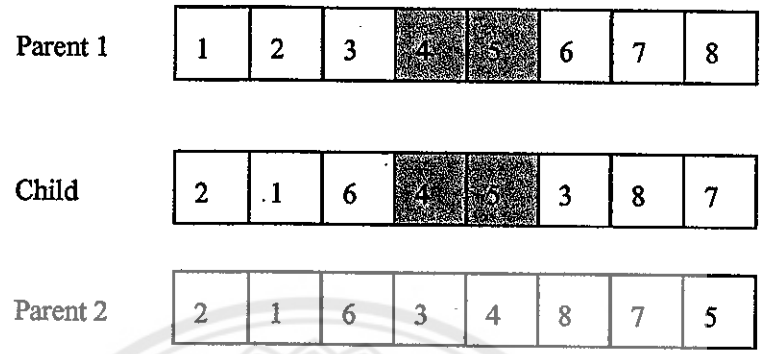
3.9.1.2 Position based crossover (Syswerda,1991)



ภาพที่ 3.11 Position based crossover (Syswerda,1991)

หลายจุดถูกเลือกอย่างสุ่มบนChromosomeของพ่อหรือแม่ แล้ววางในลูกที่ตำแหน่งเดียวกัน ส่วนที่เหลือของลูก จะสร้างจากพ่อหรือแม่ทีเหลือ โดยหลีกเลี่ยง gene ที่ซ้ำกัน กระบวนการจะถูกทำซ้ำอีกครั้งในทางตรงกันข้าม เพื่อให้ได้ลูกตัวที่ 2

3.9.1.3 Two-point centre crossover (Murata & Ishibuchi,1994)



ภาพที่ 3.12 Two-point centre crossover (Murata & Ishibuchi,1994)

หลังจาก 2 จุดถูกเลือกอย่างสุ่มๆ แล้ว ส่วนตรงกลางจะถูกคัดลอกโดยตรงไปยังลูก ส่วน gene ที่เหลือ จะถูกเติมเต็ม โดยเอาจากพ่อแม่ที่เหลือ โดยเริ่มจากซ้ายไปขวา

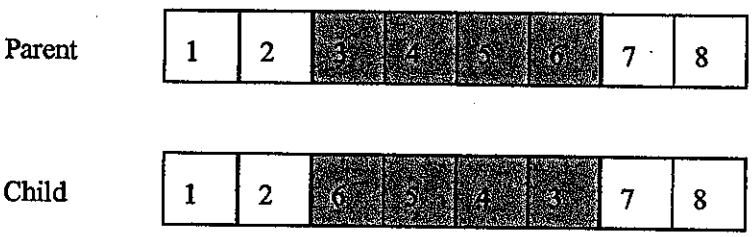
3.9.2 กระบวนการของ Mutation มีทั้งหมด 7 วิธี

3.9.2.1 Centre inverse mutation (Tralle,2000)

ภาพที่ 3.13 Centre inverse mutation (Tralle,2000)

Chromosomeถูกแบ่งเป็น 2 ส่วน gene ทั้งหมดในแต่ละส่วนถูกคัดลอกแล้ววางลงในลูก โดยป้อนกลับทางกับของเดิม โดยเริ่มจากตรงกลางวางทีละส่วน

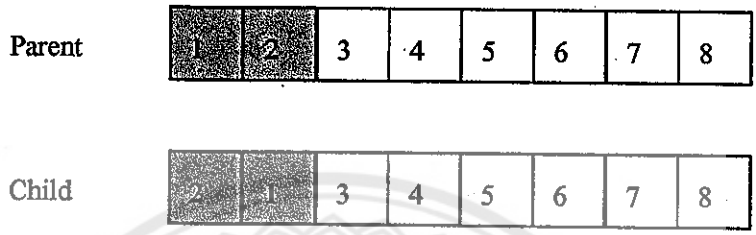
3.9.2.2 Inversion mutation (Goldberg,1989)



ภาพที่ 3.14 Inversion mutation (Goldberg,1989)

ความยาวของChromosomeย่อย จะถูกเลือกอย่างสุ่มจากพ่อหรือแม่ แล้ววางในทิศตรงข้ามในตำแหน่งเดียวกันที่ถูก

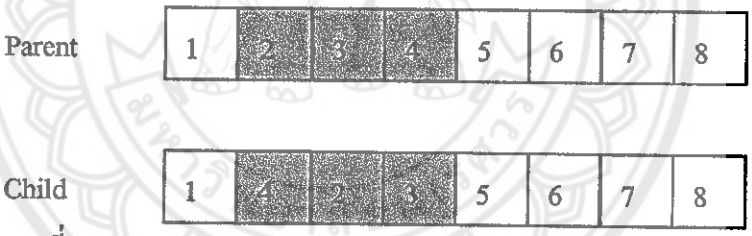
3.9.2.3 Two-operation adjacent swap (Murata & Ishibuchi,1994)



ภาพที่ 3.15 Two-operation adjacent swap (Murata & Ishibuchi,1994)

2 ตำแหน่งติดกันถูกเลือกอย่างสุ่ม 2 gene นี้จะสลับตำแหน่งกัน gene ที่เหลือคัดลอกลงมาได้เลย

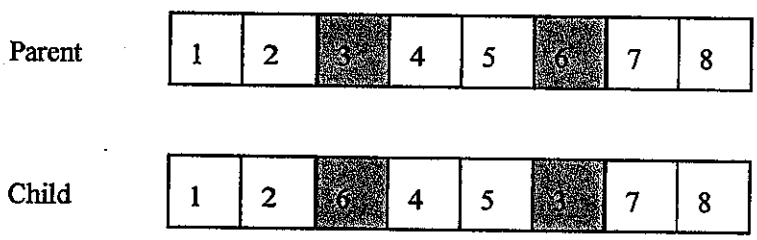
3.9.2.4 Three-operation adjacent swap (Murata & ishibuchi,1994)



ภาพที่ 3.16 Three-operation adjacent swap (Murata & ishibuchi,1994)

3 ตำแหน่งติดกันถูกเลือกอย่างสุ่มแล้วสลับเอาตัวหลังมาไว้ข้างหน้าแล้วเลื่อนตัวที่เหลือออกไปทางขวา วางเป็น gene ลูกที่ตำแหน่งเริ่มต้นเดียวกัน gene ที่เหลือจะได้จาก gene ของพ่อหรือแม่ วางลูกที่ตำแหน่งเดียวกัน

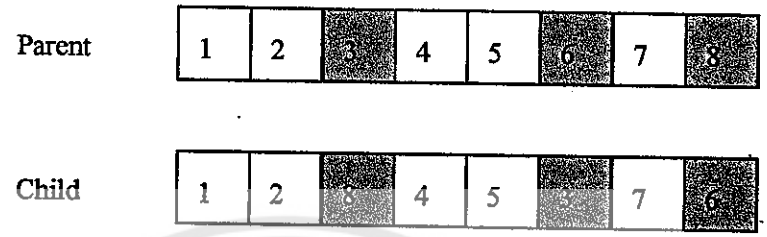
3.9.2.5 Two-operation random swap (Murata & Ishibuchi,1994)



ภาพที่ 3.17 Two-operation random swap (Murata & Ishibuchi,1994)

gene 2 ตำแหน่งที่ไม่ติดกัน ถูกเลือกอย่างสุ่ม แล้วสลับตำแหน่งกัน gene ที่เหลือคัดลอกลงมาได้เลย

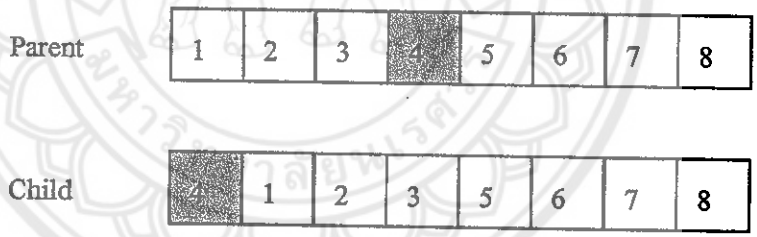
3.9.2.6 Three-operation random swap (Murata & Ishibuchi,1994)



ภาพที่ 3.18 Three-operation random swap (Murata & Ishibuchi,1994)

3 ตำแหน่งถูกเลือกอย่างสุ่ม gene เหล่านี้จะสลับตำแหน่งกัน โดยเอาตัวสุดท้ายมาไว้ที่ตำแหน่งแรกที่ถูกเลือก ตัวที่เหลือเลื่อนไปทางขวา ในตำแหน่งที่ถูกเลือกที่เหลือ ส่วน gene ที่เหลือของลูกจะถูกคัดลอกมาจาก gene ที่ไม่ถูกเลือก

3.9.2.7 Shift operation mutation (Murata & Ishibuchi,1994)



ภาพที่ 3.19 Shift operation mutation (Murata & Ishibuchi,1994)

2 ตำแหน่งถูกเลือกอย่างสุ่ม gene 2 ตำแหน่งที่เลือกมาแล้ว จะใส่ที่ตำแหน่งแรกของ gene ที่ถูกเลือก gene ที่เหลือ จะเลื่อนไปทางขวามือ

3.10 ศึกษาการเขียนโปรแกรมด้วย Visual Basic

สำหรับ Visual Basic ในปัจจุบันคือ เวอร์ชัน 6.0 ซึ่งออกมาในปี 1998 ได้เพิ่มความสามารถในการเขียนโปรแกรมติดต่อกับเครือข่ายอินเตอร์เน็ต การเชื่อมต่อกับระบบฐานข้อมูล รวมทั้งปรับปรุงเครื่องมือและการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ให้สมบูรณ์ยิ่งขึ้น พร้อมทั้งเพิ่มเครื่องมือต่างๆ อีกมากมายที่ทำให้ใช้ง่ายและสะดวกขึ้นกว่าเดิม โดยเราจะค่อยๆ มาเรียนรู้ส่วนประกอบและเครื่องมือต่างๆกัน

สาเหตุที่ Visual Basic เป็นภาษาที่เหมาะสำหรับการเรียนรู้ในการเขียนโปรแกรมนั้น เนื่องจาก Visual Basic มีข้อดีหลายประการ คือ

3.10.1 ง่ายต่อการเรียนรู้เหมาะสำหรับผู้เริ่มต้น ทั้งในเรื่องไวยากรณ์ของภาษาเองและเครื่องมือการใช้งาน ดังชื่อที่บอกอยู่แล้วว่า basic ซึ่งเหมาะสำหรับผู้เริ่มต้น

3.10.2 ความนิยมของตัวภาษา โดยอาจกล่าวได้ว่าภาษา Basic นั้นเป็นภาษาที่มีคนเรียนรู้และใช้งานมากที่สุดในประวัติศาสตร์ของคอมพิวเตอร์!

3.10.3 การพัฒนาอย่างต่อเนื่อง การปรับปรุงประสิทธิภาพในด้านของตัวภาษาและความเร็วของการประมวลผล และในเรื่องของความสามารถใหม่ๆ เช่น การติดต่อกับระบบฐานข้อมูล การเชื่อมต่อกับเครือข่ายอินเตอร์เน็ต

3.10.4 ผู้พัฒนาสำคัญของ Visual Basic คือบริษัทไมโครซอฟท์ซึ่งจัดว่าเป็นยักษ์ใหญ่ ของวงการคอมพิวเตอร์ในปัจจุบัน เราจึงสามารถมั่นใจได้ว่า Visual Basic จะยังมีการพัฒนา ปรับปรุงและคงอยู่ไปอีกนาน

นอกจาก Visual Basic มาตรฐานแล้วยังมีภาษาที่เป็นแบบเดียวกันอีก 2 แบบคือ

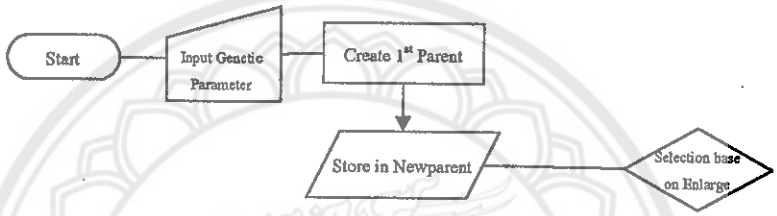
1. Visual Basic for Application Edition (VBA) ที่มาพร้อมกับชุด Microsoft Office และผลิตภัณฑ์อื่นๆ อีกมากมาย บน Windows เพื่อเพิ่มความสามารถในการเขียนโปรแกรมให้กับแอพพลิเคชันเหล่านั้น

2. VB Script Edition ที่มีการเขียนโปรแกรมเหมือนกับภาษา Visual Basic แทบทุกประการ แต่มีการเขียนเป็น Script หรือเป็นชุดคำสั่ง (คล้ายกับ Batch File ใน DOS) ในปัจจุบัน VB Script มีการใช้กันอย่างแพร่หลายในการเขียนสร้างโฮมเพจในอินเตอร์เน็ต หรือในโปรแกรมประยุกต์ที่มีการติดต่อกับระบบฐานข้อมูล และเครือข่าย

ภาษา VBA นี้จะทำให้ผู้ใช้งานสามารถใช้ภาษา Visual Basic เพื่อปรับปรุงการทำงานของโปรแกรมให้ตรงความต้องการ และมีประสิทธิภาพมากยิ่งขึ้น ตัวอย่างเช่น โปรแกรม Word ,Excel หรือ Power point ได้เตรียมภาษา VBA มาให้ผู้ใช้ ซึ่งการเขียนโปรแกรมแทบจะเหมือนกับภาษา Visual Basic ทุกประการ ทำให้ผู้ใช้สามารถปรับแต่งการทำงานของชีทคำนวณ Excel ได้หรือแม้กระทั่งเชื่อมต่อการทำงานระหว่างโปรแกรม เช่น เชื่อมข้อมูลระหว่าง Excel ,Power Point และ Word ให้ทำงานร่วมกันอย่างอัตโนมัติ ทั้ง VBA และ VB Script

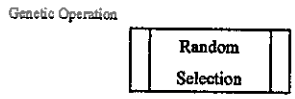
3.11 การออกแบบโปรแกรม

การอกแบบโปรแกรมนั้นเริ่มจากนำข้อมูลที่เกี่ยวข้องทั้งหมดมาพิจารณาว่าเมื่อผู้ใช้ต้องการจะคำนวณคำตอบของปัญหา โจทย์ที่จะนำเข้ามาจะมีตัวแปรใดบ้างและนำตัวแปรที่มีผลต่อกระบวนการทั้งหมดมารวมด้วยเพื่อให้ผู้ใช้สามารถกำหนดตัวแปรโจทย์ที่ต้องการได้และยังสามารถกำหนดให้โปรแกรมทำงานในรูปแบบที่ผู้ใช้ต้องการได้ และนำค่าที่ได้รับจากการป้อนเข้าของผู้ใช้มากำหนด Genetic Parameter และดำเนินงานตามขั้นตอนของกระบวนการเชิงพันธุกรรม และได้ออกแบบ Flow chart มาดังภาพที่ 3.20 ซึ่งมีคำอธิบายดังนี้



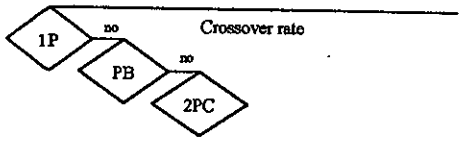
ภาพที่ 3.20

เมื่อโปรแกรมเริ่มทำงานจะรอให้ผู้ใช้ป้อนค่ากำหนดของโจทย์เข้ามาในโปรแกรมและกำหนด Parameter ที่มีโปรแกรมมีไว้ให้ตามที่ต้องการ (ผู้ใช้ไม่ต้องกำหนดก็ได้) เมื่อโปรแกรมได้รับคำสั่งให้ทำงานโปรแกรมจะสร้างต้นแบบของ Chromosome ขึ้นมาตามจำนวนของประชากร (Population size)ที่กำหนด เก็บค่าที่ได้เป็นพ่อแม่แล้วบันทึกในหน่วยความจำ ตรวจสอบว่ามีการเลือก Selection based on Enlarge sampling space หรือไม่ ถ้าใช่จะทำการสำเนาตัวพ่อแม่ส่งไปรวมกับลูกที่จะเกิดใหม่ด้วย ถ้าไม่ใช่จะไม่การทำงานในส่วนนี้



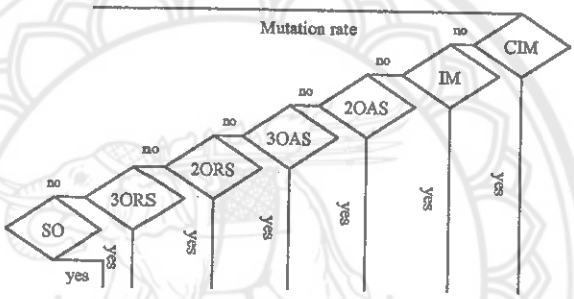
ภาพที่ 3.21

ทำการเลือกสุ่มพ่อแม่ที่จะนำไปเข้าขบวนการตัดสลับ(Crossover)ตามจำนวนอัตราการตัดสลับต่อจำนวนประชากร และสุ่มเลือกเข้าขบวนการกลายพันธุ์(Mutation)ตามจำนวนอัตราการกลายพันธุ์ต่อจำนวนประชากร และเลือกสุ่มพ่อแม่ที่จะผ่านเข้าไปยังการให้กำเนิดรอบถัดไป (Next Generation)



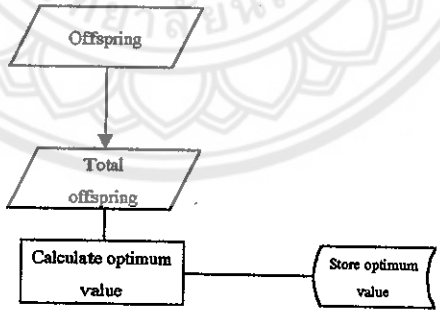
ภาพที่ 3.22

ตรวจสอบว่ามีการเลือกวิธีใดในการตัดสลับ(Crossover) และปฏิบัติตามกรรมวิธีของวิธีนั้น



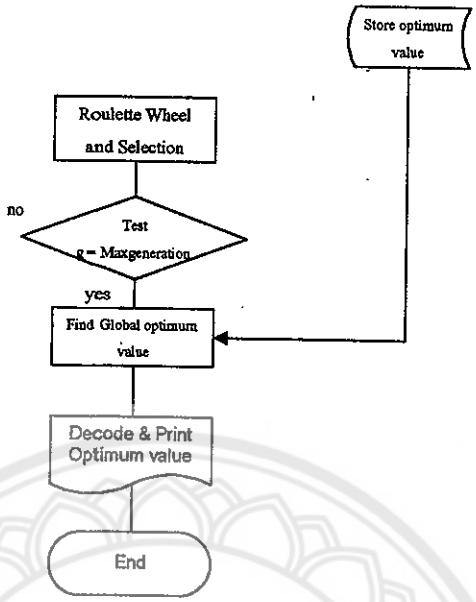
ภาพที่ 3.23

ตรวจสอบว่าเลือกวิธีใดในการกลายพันธุ์(Mutation)และปฏิบัติตามกรรมวิธีของวิธีนั้น



ภาพที่ 3.24

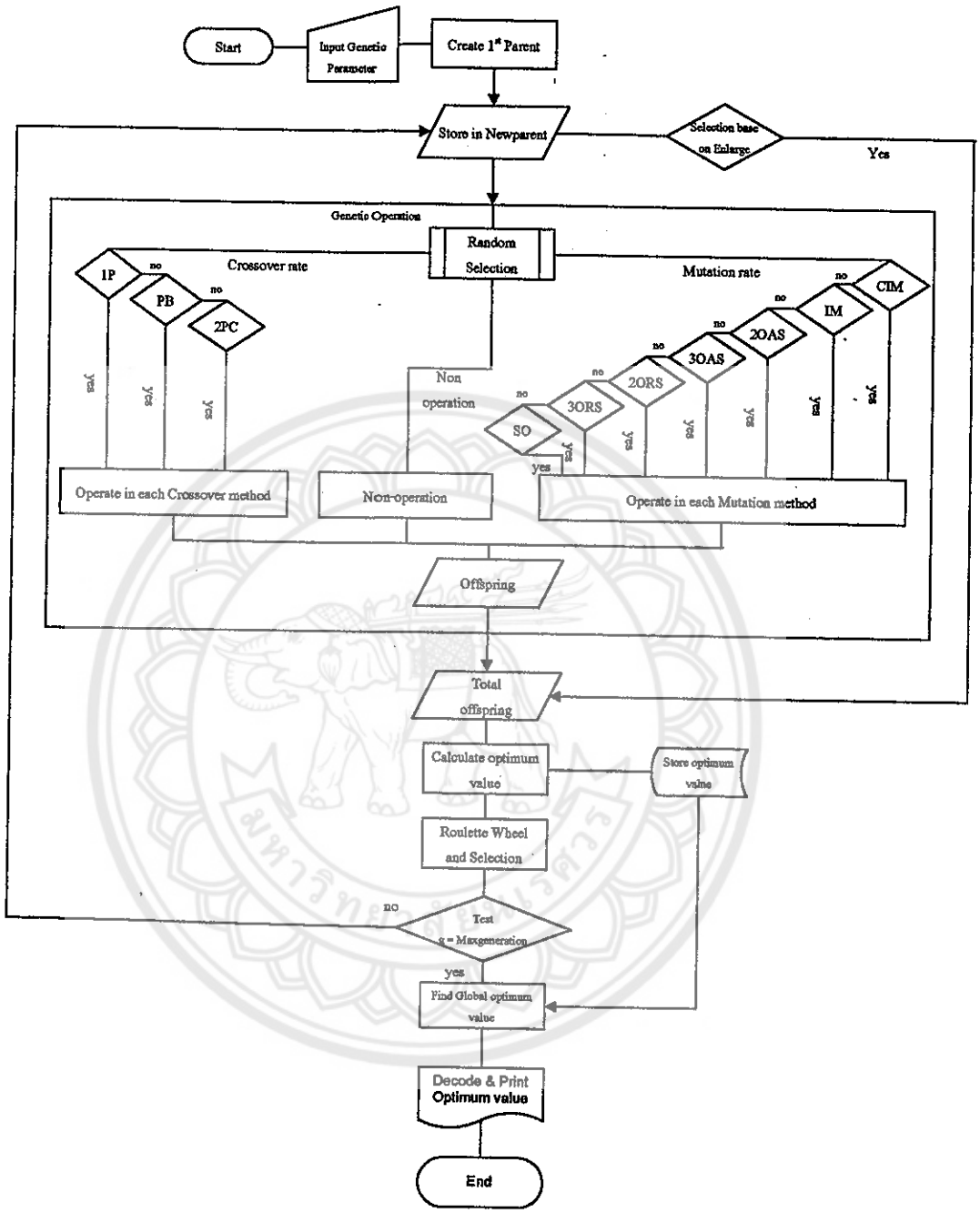
ทำการรวมลูกที่ผ่านขบวนการเชิงพันธุ์กรรมทั้งหมดและนำค่าที่ส่งเข้ามาจากพ่อแม่มาสมทบด้วย(กรณี Enlarge sampling space) จากนั้นจึงแปลงจากรูปแบบของGeneticไปเป็นรูปแบบทั่วไป และคำนวณผลลัพธ์ลูกแต่ละตัวผลลัพธ์ที่ได้นี้จะเป็นตัวกำหนดความแข็งแรงของโครโมโซมด้วย และเก็บค่าที่เหมาะสมที่สุดเอาไว้



ภาพที่ 3.25

นำค่าความแข็งแรงของโครโมโซมมากำหนดความน่าจะเป็นให้กับแต่ละโครโมโซมในขั้นตอนนี้จะทำให้โครโมโซมที่แข็งแรงมีความน่าจะเป็นมากกว่าโครโมโซมที่อ่อนแอ และสุ่มเลือกประชากรจำนวนเท่ากับ Population size เก็บค่าไว้เป็นพ่อแม่ชุดใหม่ จากนั้นตรวจสอบค่าจำนวนรอบ g ว่าเท่ากับจำนวน Generation ที่ตั้งไว้หรือเปล่าหากไม่เท่าก็ส่งพ่อแม่ชุดใหม่นี้กลับไปเข้ากรรมวิธีเชิงพันธุกรรมอีกครั้งหนึ่ง แต่หากจำนวนรอบ g ว่าเท่ากับจำนวน Generation ที่ตั้งไว้ก็จะข้ามไปกระบวนการต่อไป

ตรวจสอบค่าที่เหมาะสมที่สุด(Global optimum)จากค่าที่เหมาะสม(Local optimum)ทั้งหมดแปลงรหัสออกมาเป็นผลลัพธ์จริงและแสดงให้ผู้ใช้เห็น จากนั้นจึงจบการทำงาน



ภาพที่ 3.26 แผนภูมิแสดงขั้นตอนการดำเนินงานของโปรแกรม

3.12 การเขียนโปรแกรมและทดสอบ

งานในการเขียนโปรแกรมนี้จะถูกแบ่งออกเป็น 2 ส่วนใหญ่ๆคือ

3.12.1 ระบบของโปรแกรมโดยรวมทั้งหมด Interface ที่ใช้ติดต่อกับผู้ใช้ ฟังก์ชันการทำงานทั่วๆไป เช่น การ Save, Print ค้นหาค่าในแต่ละ Generation ได้ จัดให้มีผู้รับผิดชอบ 1 คน เนื่องจากสามารถหาคู่มือต่างๆเป็นไกด์ได้ง่าย และยังต้องทำหน้าที่เชื่อมต่อฟังก์ชันต่างๆให้ทำงานเป็นปกติ

3.12.2 Algorithm ของแต่ละวิธีเชิงพันธุกรรม (Genetic operation method) ซึ่งในโปรแกรมนี้บรรจุไว้ถึง 10 วิธีด้วยกัน แบ่งเป็นการตัดสลับ (Crossover) 3 วิธี การกลายพันธุ์ (Mutation) 7 วิธี แบ่งให้ผู้รับผิดชอบ 2 คน เนื่องจากในส่วนนี้เป็นการนเอาทฤษฎีโดยทั่วไปของ Genetic Operation มาสร้างอัลกอลิทึมในภาษา Basic ซึ่งไม่สามารถหาซื้อคู่มือมาอ้างอิงเป็นแนวทางได้จึงจำเป็นต้องใช้เวลาในการคิดและต้องมีการปรึกษาเกี่ยวกับความเข้าใจในทฤษฎีของแต่ละวิธีและแลกเปลี่ยนปัญหาและความรู้กัน

กำหนดข้อตกลงสากลระหว่างผู้รับผิดชอบทั้งหมด เช่น ตัวแปรที่ใช้ร่วมกันใช้ชื่อว่าอะไร ส่งตัวแปรออกมาเป็นชนิดใดและชื่อว่าอะไร มีการประกาศตัวแปรใดไว้บ้าง ต้องการสงวนตัวแปรใดบ้าง

เริ่มเขียนโปรแกรมในแต่ละส่วนโดยเริ่มจากการเขียนฟังก์ชั่นที่สำคัญให้ครบและเขียนวิธีการกลายพันธุ์และวิธีการตัดสลับอย่างละ 1 วิธี เพื่อที่จะสามารถประกอบเป็นโปรแกรมที่สามารถคำนวณค่าได้ครบเต็มรอบและทดสอบความเข้ากันจากการประกอบกันของฟังก์ชันทั้งหมดซึ่งเขียนจาก 3 คน เมื่อพบข้อผิดพลาดจะร่วมกันวิเคราะห์หาจุดเกิดเหตุและช่วยกันแก้ไข

ทดสอบความถูกต้องของโปรแกรมโดยการรันโปรแกรมที่ Population sizeน้อยๆแล้วจึงคำนวณค่าจากวิธีอื่นมาเปรียบเทียบ หากมีข้อผิดพลาดจะตรวจสอบทีละ 1 ฟังก์ชันเพื่อหาฟังก์ชันที่ผิดพลาดและให้ผู้ที่เขียนนั้นอธิบายโค้ดให้คนที่เหลือฟังโดยละเอียดเพื่อเป็นข้อมูลในการวิเคราะห์และแก้ไข เมื่อแก้ไขเสร็จแล้วจึงเก็บเป็น Demo

เริ่มเขียนฟังก์ชันอื่นเพิ่มเติมโดยปกติแล้วจะไม่เขียนเพิ่มใน Demo เลย แต่จะใช้การเขียนเป็น Project ใหม่เมื่อทดสอบแล้วว่าฟังก์ชั่นนั้นทำงานเพียงตัวเดียวถูกต้องจึงค่อยนำมาประกอบกับ Demo ที่เขียนไว้ แล้วทำการทดสอบความเข้ากันได้และความถูกต้องอีกครั้งหนึ่ง

3.13 การพัฒนาและแก้ไขโปรแกรม

ในการพัฒนาและแก้ไขโปรแกรมนั้นจะใช้คำแนะนำจากอาจารย์ที่ปรึกษาเป็นหลักเนื่องจาก Genetic Algorithm นี้เป็นความรู้ในสมัยใหม่ยังไม่แพร่หลายทั่วไปมากนักต้องค้นคว้าเพิ่มเติมเท่านั้นจึงหาผู้มีประสบการณ์ได้น้อย จึงให้อาจารย์ที่ปรึกษาเป็นผู้ให้คำติชมและนำไปปรับปรุงโปรแกรมและนำมาเสนอใหม่

3.14 สมมติฐานและขอบเขตของโปรแกรม

โปรแกรมชุดนี้ได้จับกระบวนการเชิงพันธุกรรมมาใช้แก้ปัญหาในรูปแบบปัญหาของการเดินทางของพนักงานขาย(Traveling Salesman Problem) และใช้ขอบเขตของการเดินทั่วทุกจังหวัดในประเทศไทยเป็นกรณีศึกษา ซึ่งจะมีสมมติฐานดังนี้

การเดินทางระหว่างจังหวัดนั้นจะคิดระยะ ทางจากจังหวัดหนึ่งถึงอีกจังหวัดหนึ่งเท่านั้นจะไม่คิดรวมกันกับระยะทางที่เดินทางภายในจังหวัด

ข้อมูลที่นำมาใช้อ้างอิงนี้นำมาจากฐานข้อมูลของโปรแกรมคำนวณระยะทางระหว่างจังหวัดใน Website ของกรมทางหลวง ซึ่งต้องนำมาถอดรหัสจากไฟล์.html แล้วนำมาบรรจุเป็นฐานข้อมูลรวมในไฟล์ของ Ms-Access ซึ่งข้อมูลของกรมทางหลวงจะแบ่งแยกข้อมูลออกเป็นส่วนย่อยๆ โดยมีกรุงเทพฯเป็นจุดเชื่อมต่อ ในการรวมตารางทั้งหมดเข้าด้วยกันนั้นจำเป็นจะต้องใช้การบวกเส้นทางภายในส่วนย่อยนั้นผ่านกรุงเทพเสียก่อน เมื่อตรวจสอบผลลัพธ์ที่ได้นั้นพบว่ามีค่าเท่ากันกับผลลัพธ์ของกรมทางหลวงในทุกๆค่าที่ตรวจสอบ

ขอบเขตความสามารถของโปรแกรม

3.14.1 กำหนดจำนวนประชากร(Population size) ได้ตั้งแต่ 5 ถึง 100

3.14.2 กำหนดจำนวนการให้กำเนิดใหม่(Number of Generation) ได้ตั้งแต่ 1 ถึง 1000

3.14.3 สามารถกำหนดจำนวนยีน(Number of gene) ได้ตั้งแต่ 3 ถึง 100

3.14.4 สามารถกำหนดอัตราตัดสลับ (Crossover rate) ได้ตั้งแต่ 0% ถึง 100%

3.14.5 สามารถกำหนดอัตราการกลายพันธุ์ (Mutation rate) ได้ตั้งแต่ 0% ถึง 100%

3.14.6 เลือกกำหนด Optimize function เป็น Minimize หรือ Maximize ได้

3.14.7 สามารถกำหนดชุดเลขสุ่ม (Seed) เองได้ตั้งแต่ 1ถึง 100

3.14.8 กำหนดฐานการสรรหาได้ 2 รูปแบบคือ Regular sampling space และ Enlarge sampling space

3.14.9 สามารถกำหนดรูปแบบการเดินทางได้ 2 แบบคือ Close loop และ Open loop ซึ่งแต่ละวิธียังสามารถกำหนดจุดเริ่มต้นและจุดสุดท้ายได้

3.14.10 เลือกวิธีในการกลายพันธุ์ (Mutation) ได้ 7 วิธี

3.14.11 เลือกวิธีการตัดสลับ (Crossover) ได้ 3 วิธี ซึ่งสามารถกำหนดได้อย่างอิสระจากวิธีการกลายพันธุ์

3.14.12 สามารถเรียกดูผลลัพธ์ได้ทุกๆ โครโมโซมโดยเพียงป้อนค่า Generation และตำแหน่งของโครโมโซมเท่านั้น

3.14.13 เมื่อคำนวณครบกระบวนการแล้วสามารถนำประชากรชุดสุดท้ายไปเริ่มกระบวนการGenetic ต่อไปได้ทันที

3.14.14 สามารถจัดเก็บผลลัพธ์และค่าต่างๆที่เกี่ยวข้องจากการคำนวณหนึ่งรอบในรูปของ Text Files (*.txt)

3.14.15 สามารถแสดงผลผ่านทางเครื่องพิมพ์ได้

มหาวิทยาลัยนเรศวร

บทที่ 4
การวิจัยและผลของการวิจัย

4.1 การทดสอบการทำงานของโปรแกรม

จากทฤษฎีโดยทั่วๆไปของ Genetic Algorithm นั้นจะมีกลไกความอยู่รอดที่จะคัดสรรเอา Chromosome ที่แข็งแรงกว่าอยู่รอดต่อไปในช่วงอายุถัดไป (Next Generation) ซึ่ง Chromosome ที่แข็งแรงหมายถึงมีค่าคำตอบของฟังก์ชันที่ดี ดังนั้นถ้าจำนวนช่วงอายุมีค่ามากๆแล้วจะทำให้มีโอกาสได้พบกับคำตอบของสมการที่ดีด้วย และจากทฤษฎีนี้ถ้าหากเรากำหนด Number of Generation(จำนวนช่วงอายุ)ในโปรแกรมให้มากขึ้นก็ย่อมจะต้องทำให้คำตอบที่พบนั้นดีขึ้นด้วย ดังนั้นจึงออกแบบการทดสอบศักยภาพของโปรแกรมว่าสามารถทำงานได้ตามทฤษฎีทั่วไปหรือไม่ จึงได้กำหนด Number of Generation(ช่วงอายุ) = 10, 25, 50, 100 แล้วทำงานเปรียบเทียบกัน แต่เนื่องจากยังมีตัวแปรอีกหลายตัวที่ส่งผลต่อการหาคำตอบหากทำการทดลองทั้งหมดนั้นจะต้องใช้เวลามาก ดังนั้นจึงนำข้อมูลจากผลงานวิจัยทางวิชาการที่เกี่ยวข้องและทำการศึกษาปัจจัยที่เกี่ยวข้องกับการหาคำตอบที่ดี มาอ้างอ้างอิงค่าตัวแปรต่างๆดังนี้

- Genetic operation method
  - > Position-based Crossover (P.Pongcharoen 2001)
  - > Two Operation Adjacent Swap Mutation (P.Pongcharoen 2001)
- Selection based on regular sampling space (P.Pongcharoen 2001)
- 90% Crossover rate, 18% Mutation rate (P.Pongcharoen 2001)

และตัวแปรอื่นนั้นต้องปรับให้เหมาะสมกับกรณีศึกษาของงานวิจัยนี้โดยกำหนดตัวแปรดังนี้

- Population size = 100
- Number of gene = 76
- Minimize function
- Close loop, Started and finished at 2(กรุงเทพฯ)
- Random Seed ครั้งแรก และกำหนดให้เหมือนกันในครั้งต่อไป

4.2 ผลการทดสอบ

ทำการรันโปรแกรมตามข้อกำหนดทั้งหมด 4 รอบๆละ 4 ครั้งแต่ละครั้งกำหนด Number of Generation ที่ต่างกันได้ผลดังนี้

รอบที่ 1 ครั้งที่ 1 Number of Generation = 10

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 10

Minimize function Seed is : 69 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 46159

Traveling Route :

กรุงเทพ อำนาจเจริญ ตาก ระยอง ลำปาง สุโขทัย อ่างทอง กาจนบุรี สระแก้ว เพชรบูรณ์ เชียงราย มุกดาหาร อยุธยา สระบุรี สิงห์บุรี ร้อยเอ็ด หนองคาย ปทุมธานี สตูล พิจิตร เชียงใหม่ อุดรธานี สมุทรปราการ กระบี่ นครราชสีมา อุบลราชธานี นนทบุรี ประจวบคีรีขันธ์ พังงา หนองบัวลำภู มหาสารคาม เลย ตรัง นราธิวาส ยะลา สมุทรสาคร นครปฐม กำแพงเพชร ปราจีนบุรี นครนายก ชัยนาท นครศรีธรรมราช ลพบุรี ชุมพร ศรีสะเกษ อุทัยธานี บุรีรัมย์ กาฬสินธุ์ พะเยา อุตรดิตถ์ ฉะเชิงเทรา แม่ฮ่องสอน สงขลา ตราด เพชรบุรี สมุทรสงคราม ระนอง ขอนแก่น สุรินทร์ ยโสธร แพร่ จันทบุรี สกลนคร ปัตตานี สุราษฎร์ธานี นครพนม สุพรรณบุรี พิษณุโลก ลำพูน น่าน ชลบุรี พัทลุง ภูเก็ต ราชบุรี นครสวรรค์ ชัยภูมิ กรุงเทพ

รอบที่ 1 ครั้งที่ 2 Number of Generation = 25

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 25

Minimize function Seed is : 69 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 45052

Traveling Route :

กรุงเทพ อำนาจเจริญ ระยอง ชลบุรี ตาก ลำปาง สุโขทัย ปทุมธานี กาจนบุรี อ่างทอง เพชรบูรณ์ สระแก้ว เชียงราย มุกดาหาร หนองคาย สระบุรี ร้อยเอ็ด พิจิตร เชียงใหม่ อุดรธานี กาฬสินธุ์ อุบลราชธานี สมุทรปราการ ประจวบคีรีขันธ์ กระบี่ นครศรีธรรมราช ชัยภูมิ อุทัยธานี พังงา ระนอง นครราชสีมา สตูล หนองบัวลำภู มหาสารคาม เลย ปราจีนบุรี

นครนายก ชัยนาท นราธิวาส ยะลา ชุมพร อยุธยา สมุทรสาคร ตรัง แพร่ นครปฐม กำแพงเพชร นนทบุรี ลพบุรี อุตรดิตถ์ ศรีสะเกษ แม่ฮ่องสอน บุรีรัมย์ พะเยา ฉะเชิงเทรา เพชรบุรี สงขลา ตราด สุรินทร์ สมุทรสงคราม ขอนแก่น ยโสธร นครพนม จันทบุรี สุราษฎร์ธานี ปัตตานี สกลนคร สุพรรณบุรี พิษณุโลก น่าน ลำพูน ราชบุรี พัทลุง ภูเก็ต สิงห์บุรี นครสวรรค์ กรุงเทพ

รอบที่ 1 ครั้งที่ 3 Number of Generation = 50

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 50

Minimize function Seed is : 69 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 44529

Traveling Route :

กรุงเทพ อำนาจเจริญ อุบลราชธานี ระยอง ลำปาง อ่างทอง เชียงใหม่ สุโขทัย กาจนบุรี เชียงราย สระแก้ว สระบุรี มุกดาหาร ปทุมธานี ร้อยเอ็ด ยโสธร นนทบุรี หนองคาย อุดรธานี นครราชสีมา ประจวบคีรีขันธ์ พังงา สตูล กาฬสินธุ์ ชัยภูมิ น่าน กระบี่ นครศรีธรรมราช ระนอง เลย มหาสารคาม หนองบัวลำภู นครนายก สมุทรปราการ ตาก ยะลาสมุทรสาคร นราธิวาส ตรัง นครปฐม เพชรบูรณ์ กำแพงเพชร ศรีสะเกษ ปราจีนบุรี ชัยนาท อยุธยา ลพบุรี ชุมพร พิจิตร แพร่ อุตรดิตถ์ แม่ฮ่องสอน

อุทัยธานี บุรีรัมย์ ฉะเชิงเทรา ตราด สงขลา เพชรบุรี ขอนแก่น นครพนม สมุทรสงคราม สุรินทร์ สกลนคร สุราษฎร์ธานี พะเยา สุพรรณบุรี จันทบุรี ปัตตานี ลำพูน พิษณุโลก ชลบุรี ภูเก็ต พัทลุง สิงห์บุรี นครสวรรค์ ราชบุรี กรุงเทพ

รอบที่ 1 ครั้งที่ 4 Number of Generation = 100

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 100

Minimize function Seed is : 69 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 44529

Traveling Route :

กรุงเทพ อำนาจเจริญ อุบลราชธานี ระยอง ลำปาง อ่างทอง เชียงใหม่ สุโขทัย กาจนบุรี เชียงราย สระแก้ว สระบุรี มุกดาหาร ปทุมธานี ร้อยเอ็ด ยโสธร นนทบุรี หนองคาย อุดรธานี นครราชสีมา ประจวบคีรีขันธ์ พังงา สตูล กาฬสินธุ์ ชัยภูมิ น่าน กระบี่ นครศรีธรรมราช ระนอง เลย มหาสารคาม หนองบัวลำภู นครนายก สมุทรปราการ ตาก ยะลา สมุทรสาคร นราธิวาส ตรัง นครปฐม เพชรบูรณ์ กำแพงเพชร ศรีสะเกษ ปราจีนบุรี ชัยนาท อยุธยา ลพบุรี ชุมพร พิจิตร แพร่ อุตรดิตถ์ แม่ฮ่องสอน อุทัยธานี บุรีรัมย์ ฉะเชิงเทรา ตราด สงขลา เพชรบุรี ขอนแก่น นครพนม สมุทรสงคราม สุรินทร์ สกลนคร สุราษฎร์ธานี พะเยา สุพรรณบุรี จันทบุรี ปัตตานี ลำพูน พิษณุโลก ชลบุรี ภูเก็ต พัทลุง สิงห์บุรี นครสวรรค์ ราชบุรี กรุงเทพ

ตารางที่ 4.1 ข้อมูลตั้งแต่ Generation ที่ 1 ถึง 100 (Seed 69)

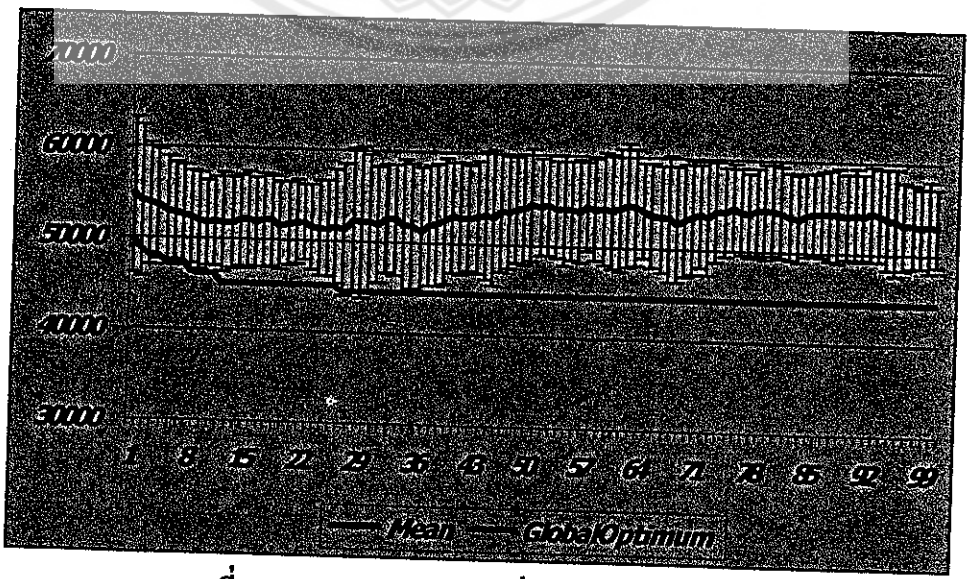
Generation	Local Optimum	Mean	SD	Global Optimum
1	49804	54701.94	2837.32	49804
2	48732	54126.22	2767.94	48732
3	48060	53769.76	2246.54	48060
4	48572	53663.62	2019.99	48060
5	47397	53021.29	2003.11	47397
6	48211	52845.49	1836.46	47397
7	48133	52587.49	1862.58	47397
8	46348	52408.12	1712.26	46348
9	46159	51921.64	1745.24	46159
10	47042	51656.9	1576.27	46159
11	46458	51894.34	1607.2	46159
12	45052	51742.39	1819.22	45052
13	46018	51772.05	1652.6	45052
14	46344	52022.66	1687.67	45052
15	46559	52137.49	1777.68	45052
16	46345	51964.14	1648.63	45052
17	47014	52143.92	1702.58	45052
18	46358	52070.82	1655.84	45052
19	46820	51509.07	1593.78	45052
20	47336	51756.28	1541.91	45052
21	47107	51965.08	1512.62	45052
22	46571	51991.36	1403.13	45052
23	46589	51474.87	1577.3	45052
24	46589	51347.36	1757.58	45052
25	45945	51366.42	1753.78	45052
26	45177	51585.56	2068.03	45052
27	44529	51221.61	2418.99	44529

Generation	Local Optimum	Mean	SD	Global Optimum
28	45188	52299.72	2626.42	44529
29	44913	52191.75	2778.36	44529
30	45163	52121.99	2482.75	44529
31	45982	51959.71	2137.91	44529
32	46388	52491.84	1973.6	44529
33	47376	52447.7	1988.99	44529
34	46454	51944	2403	44529
35	46643	51748.3	2293.74	44529
36	46870	51380.85	2282.21	44529
37	46606	51896.63	2171.59	44529
38	45136	52028.94	2255.35	44529
39	47088	52508.91	2312.22	44529
40	46800	52904.99	2027.61	44529
41	45582	52723.81	2036.92	44529
42	44839	52805.68	1951.96	44529
43	45661	53007.05	2052.46	44529
44	45953	53195.32	2312.7	44529
45	46541	53006.82	2494.82	44529
46	48187	53566.88	2155.65	44529
47	47737	53619.16	2052.59	44529
48	47627	53915.32	1935.36	44529
49	47974	53960.53	2026.13	44529
50	47684	54391.83	2051.23	44529
51	48962	54210.82	1770.08	44529
52	50100	54468.64	1833.37	44529
53	50242	54064.33	1769.08	44529
54	49648	53961.18	2008.1	44529
55	49665	53937.11	1828.05	44529

Generation	Local Optimum	Mean	SD	Global Optimum
56	49024	53889.73	1878.85	44529
57	50353	54447.19	1630.37	44529
58	49222	54144.5	1867.76	44529
59	48475	54211.8	2077.52	44529
60	48217	53964.36	2088.76	44529
61	48039	54156.5	2204.58	44529
62	48764	54714.04	2263.12	44529
63	49056	54434.19	2136.3	44529
64	49013	54111.95	1757.09	44529
65	48183	53572.59	1934.31	44529
66	48052	53398.97	1951.3	44529
67	48659	53382.11	2310.39	44529
68	48280	52785.3	2128.48	44529
69	48767	53214.87	2083.95	44529
70	47192	53533.27	2106	44529
71	49229	53726.87	1895.3	44529
72	46652	53513.2	2059.44	44529
73	46644	54233.22	1857.07	44529
74	50076	54234.86	1632.41	44529
75	49403	54748.9	1597.53	44529
76	50045	54292.26	1588.51	44529
77	49913	53945.41	1550.23	44529
78	49531	54384.55	1606.36	44529
79	48159	54372.27	1730.98	44529
80	47472	54560.69	1688.09	44529
81	48152	54327.62	1673.89	44529
82	49883	53915.8	1597.02	44529
83	48988	53711.65	1616.3	44529

Generation	Local Optimum	Mean	SD	Global Optimum
84	49124	54107.24	1540.53	44529
85	49104	54281.22	1478.42	44529
86	50403	54220.88	1558.19	44529
87	48445	54215.94	1660.31	44529
88	49074	54170.34	1714.69	44529
89	48391	54102.64	1566.12	44529
90	48542	54244.7	1677.31	44529
91	48452	54085.63	1575.44	44529
92	47976	54373.16	1729.01	44529
93	48680	54303.64	1695.03	44529
94	47993	53817.26	1867.73	44529
95	47338	53518.41	1962.61	44529
96	47102	53118.67	1602.8	44529
97	48055	53124.98	1535.9	44529
98	47005	52967.38	1457.1	44529
99	46691	53108.95	1658.47	44529
100	48245	53010.61	1424.54	44529

นำผลลัพธ์ทั้ง 100 Generation ไปเขียนเป็นกราฟความสัมพันธ์



ภาพที่ 4.1 กราฟแสดงความเปลี่ยนแปลงใน 100 Generation

วิเคราะห์ผลการทดลอง

ค่า Global Optimum เป็นค่าที่ต่ำที่สุดในแต่ละ Generation จากรูปเมื่อ Generation มีค่าเพิ่มขึ้นค่าที่ดีที่สุดก็ลดลงด้วย และจากการสังเกตค่า Global optimum จะลดลงเมื่ออยู่ในช่วงการกระจายตัวของข้อมูลมากๆ

รอบที่ 2 ครั้งที่ 1 Number of Generation = 10

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 10

Minimize function Seed is : 2 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 46566

Traveling Route :

กรุงเทพ สงขลา นราธิวาส อ่างทอง สกลนคร ชลบุรี อยุธยา สตูล กระบี่ ตราด สุรินทร์ ปทุมธานี หนองคาย กำแพงเพชร สุราษฎร์ธานี เพชรบุรี ชุมพร สระบุรี ระยอง นครปฐม สุโขทัย ปัตตานี ราชบุรี พิจิตร พิษณุโลก ปราจีนบุรี ระนอง ประจวบคีรีขันธ์ อุดรธานี หนองบัวลำภู นครศรีธรรมราช ลพบุรี เชียงใหม่ ชัยนาท ศรีสะเกษ นครสวรรค์ จันทบุรี สระแก้ว กาจนบุรี สมุทรปราการ พังงา อุตรดิตถ์ สมุทรสงคราม ภูเก็ต ตรัง มุกดาหาร นครนายก พะเยา อำนาจเจริญ บุรีรัมย์ นครพนม ชัยภูมิ ลำปาง แพร่ พัทลุง สมุทรสาคร เลย สุพรรณบุรี ตาก ร้อยเอ็ด ขอนแก่น ฉะเชิงเทรา อุทัยธานี กาฬสินธุ์ ลำพูน น่าน มหาสารคาม ยะลา นครราชสีมา ยโสธร อุบลราชธานี สิงห์บุรี เพชรบูรณ์ แม่ฮ่องสอน เชียงราย นนทบุรี กรุงเทพ

รอบที่ 2 ครั้งที่ 2 Number of Generation = 25

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 25

Minimize function Seed is : 2 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 46566

Traveling Route :

กรุงเทพ สงขลา นราธิวาส อ่างทอง สกลนคร ชลบุรี อยุธยา สตูล กระบี่ ตราด สุรินทร์ ปทุมธานี หนองคาย กำแพงเพชร สุราษฎร์ธานี เพชรบุรี ชุมพร สระบุรี ระยอง นครปฐม สุโขทัย ปัตตานี ราชบุรี พิจิตรพิษณุโลก ปราจีนบุรี ระนอง ประจวบคีรีขันธ์ อุดรธานี หนองบัวลำภู นครศรีธรรมราช ลพบุรี เชียงใหม่ ชัยนาท ศรีสะเกษ นครสวรรค์ จันทบุรี สระแก้ว กาจนบุรี สมุทรปราการ พังงา อุตรดิตถ์ สมุทรสงคราม ภูเก็ต ตรัง มุกดาหาร นครนายก พะเยา อำนาจเจริญ บุรีรัมย์ นครพนม ชัยภูมิ ลำปาง แพร่ พัทลุง สมุทรสาคร เลย สุพรรณบุรี ตาก ร้อยเอ็ด ขอนแก่น ฉะเชิงเทรา อุทัยธานี กาฬสินธุ์ ลำพูน น่าน มหาสารคาม ยะลา นครราชสีมา ยโสธร อุบลราชธานี สิงห์บุรี เพชรบูรณ์ แม่ฮ่องสอน เชียงราย นนทบุรี กรุงเทพ

รอบที่ 2 ครั้งที่ 3 Number of Generation = 50

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 50

Minimize function Seed is : 2 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 45146

Traveling Route :

กรุงเทพ ประจวบคีรีขันธ์ สุรินทร์ ระยอง ปทุมธานี ชลบุรี แม่ฮ่องสอน หนองคาย เพชรบุรี นราธิวาส มหาสารคาม สุโขทัย สระบุรี หนองบัวลำภู นครพนม สระแก้ว มุกดาหาร ขอนแก่น สกลนคร ปราจีนบุรี พิจิตร พิษณุโลก ลำปาง ลำพูน ยโสธร อุดรธานี สมุทรสงคราม ระนอง ยะลา สงขลา นครศรีธรรมราช อยุธยา นครนายก ปัตตานี ภูเก็ต น่าน เชียงใหม่ ชัยนาท ลพบุรี กาฬสินธุ์

ตราด นครราชสีมา เชียงราย กระบี่ ศรีสะเกษ ราชบุรี อุทัยธานี พังงา กาจนบุรี อุตรดิตถ์ กำแพงเพชร พะเยา สตูล สุราษฎร์ธานี ตรัง อำนาจเจริญ บุรีรัมย์ นครปฐม พัทลุง ร้อยเอ็ด นครสวรรค์ ชัยภูมิ อุบลราชธานี ฉะเชิงเทรา จันทบุรี สิงห์บุรี สุพรรณบุรี สมุทรปราการ อ่างทอง นนทบุรี เลย แพร่ สมุทรสาคร เพชรบูรณ์ ตาก ชุมพร กรุงเทพ

รอบที่ 2 ครั้งที่ 4 Number of Generation = 100

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 100

Minimize function Seed is : 2 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 44664

Traveling Route :

กรุงเทพ ระยอง สุรินทร์ ปทุมธานี นราธิวาส สระแก้ว ประจวบคีรีขันธ์ หนองคาย เพชรบุรี มหาสารคาม สระบุรี ปราจีนบุรี มุกดาหาร ลำพูน ระนอง บุรีรัมย์ นครนายก พิจิตร ปัตตานี นครศรีธรรมราช สมุทรสงคราม นครพนม ยโสธร สุโขทัย หนองบัวลำภู อยุธยา นครราชสีมา อุดรธานี เชียงใหม่ ลำปาง ลพบุรี สงขลา กระบี่ ราชบุรี ชัยนาท ชลบุรี อุตรดิตถ์ กาฬสินธุ์ ศรีสะเกษ อุทัยธานี ตราด นครสวรรค์ ขอนแก่น กำแพงเพชร เชียงราย พะเยา น่าน กาจนบุรี พังงา สตูล ชุมพร อำนาจเจริญ สุราษฎร์ธานี พิษณุโลก สกลนคร ร้อยเอ็ด อุบลราชธานี ชัยภูมิ นครปฐม สุพรรณบุรี ฉะเชิงเทรา สมุทรปราการ สิงห์บุรี พัทลุง ยะลา จันทบุรี อ่างทอง นนทบุรี แพร่ เลย สมุทรสาคร ภูเก็ต แม่ฮ่องสอน เพชรบูรณ์ ตรัง ตาก กรุงเทพ

ตารางที่ 4.2 ข้อมูลตั้งแต่ Generation ที่ 1 ถึง 100 (Seed 2)

Generation	Local Optimum	Mean	SD	Global Optimum
1	48956	54297.15	2727.36	48956
2	48728	54049.45	2303.32	48728
3	46923	53699.88	2468.01	46923
4	47296	53643.3	2332.45	46923

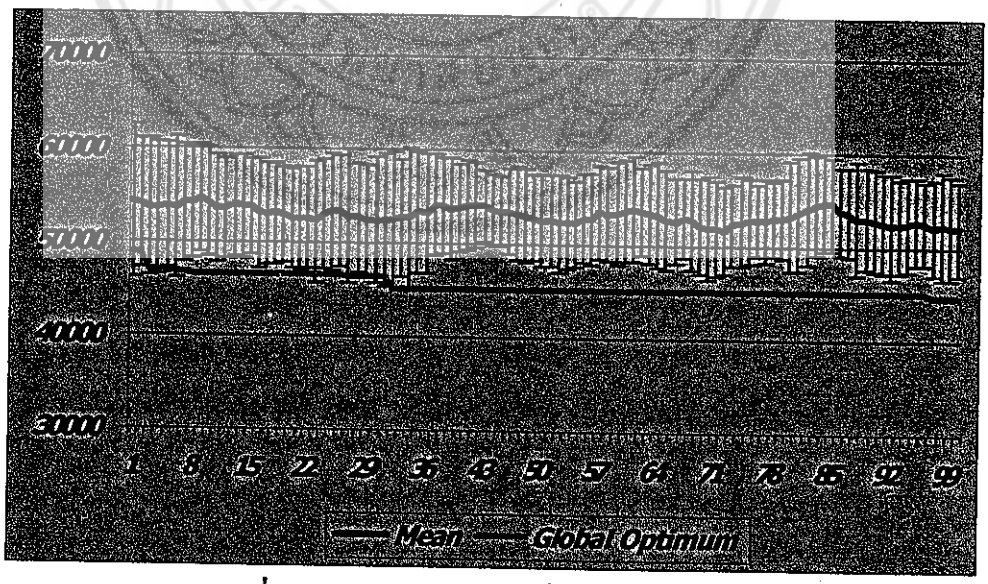
Generation	Local Optimum	Mean	SD	Global Optimum
5	46718	53894.51	2240.59	46718
6	47708	53704.67	2501.01	46718
7	46990	54215.58	2096.21	46718
8	46566	54186	1983.66	46566
9	46681	54633.58	1931.04	46566
10	48935	54347.26	1972.55	46566
11	47957	53633.62	1906.56	46566
12	47845	53626.1	1777.28	46566
13	49208	53992.41	1804.33	46566
14	48946	53846.34	1817.2	46566
15	48852	53758.11	1703.82	46566
16	47731	53100.45	1965.32	46566
17	46982	53069.57	1830.25	46566
18	48015	53100.86	1833.29	46566
19	47338	52956.89	1718.42	46566
20	47373	52581.01	1790.05	46566
21	47609	52568.6	1850.22	46566
22	47650	52619.15	1830.46	46566
23	47070	52289.83	2152.84	46566
24	48219	53020.95	1946.99	46566
25	46846	53304.59	2093.91	46566
26	46616	53456.45	1998.82	46566
27	46437	52856.33	2404.62	46437
28	47313	52701.12	1964.43	46437
29	46401	52369.41	1944.59	46401
30	47149	52422.67	2252.1	46401
31	47128	52444.56	2344.19	46401
32	45146	52615.04	2424.96	45146

Generation	Local Optimum	Mean	SD	Global Optimum
33	45323	52729.14	2090.77	45146
34	45250	52729.53	2528.44	45146
35	46474	53339.94	2183.37	45146
36	47634	53283.19	2123.42	45146
37	47465	54129.14	1784.35	45146
38	49595	54020.23	1680.99	45146
39	48381	53708.14	1802.1	45146
40	48381	53762.09	1560.21	45146
41	48716	54148.39	1581.59	45146
42	48366	54079.81	1523.44	45146
43	49033	54021.36	1275.59	45146
44	49138	53566.44	1271.84	45146
45	48747	53454.53	1248.05	45146
46	49015	53478.89	1528.61	45146
47	47311	53244.24	1656.9	45146
48	47751	53002.14	1659.79	45146
49	47639	52856.88	1521.05	45146
50	47339	52567.7	1627.55	45146
51	47718	52586.51	1628.55	45146
52	48005	52257.38	1735.94	45146
53	47263	52302.51	1629.15	45146
54	47488	52242.55	1512.06	45146
55	48415	52777.74	1595.27	45146
56	48284	53546.51	1645.37	45146
57	48056	52721.83	1618.13	45146
58	49093	53269.78	1666.58	45146
59	48222	53406.47	1826.51	45146
60	49023	53722.88	1835.27	45146

Generation	Local Optimum	Mean	SD	Global Optimum
61	49614	53932.16	1845.4	45146
61	49614	53932.16	1845.4	45146
62	49281	53634.83	1564.22	45146
63	47347	53377.62	1786.99	45146
64	47405	53288.23	1893.43	45146
65	48908	52805.44	1753.55	45146
66	47561	52613.9	1516.73	45146
67	46257	52968.31	1433.47	45146
68	45549	52698.21	1779.11	45146
69	47137	52157.31	1724.23	45146
70	46794	51910.49	1831.43	45146
71	47068	51934.44	1727.11	45146
72	47068	51687.14	1573.76	45146
73	48143	52241.5	1441.16	45146
74	48161	52623.52	1573.23	45146
75	48166	52697.4	1542.53	45146
76	49072	52783.05	1260.74	45146
77	48149	52706.12	1485.49	45146
78	49332	53003.7	1341.31	45146
79	47565	52939.16	1435.51	45146
80	48218	53292.72	1960.89	45146
81	46939	53538.1	2027.67	45146
82	47895	54148.53	2000.49	45146
83	49088	54525.36	1780.27	45146
84	49088	54727.96	1733.88	45146
85	48915	54617.11	1683.25	45146
86	48313	53754.07	1564.99	45146
87	47424	53704.75	1624.88	45146

Generation	Local Optimum	Mean	SD	Global Optimum
88	46545	53072.07	1999.31	45146
89	48493	53048.27	1787.89	45146
90	47302	52795.26	1807.57	45146
91	48066	52529.08	1771.76	45146
92	48613	52268.1	1902.62	45146
93	48583	52280.97	1657.17	45146
94	46845	52497.27	1678.59	45146
95	46646	52734.7	1544.66	45146
96	46126	52604.56	1485.04	45146
97	44664	52071.52	1690.62	44664
98	46471	52364.46	1916.16	44664
99	47658	52111.12	1867.74	44664
100	48323	52031.85	1658.91	44664

นำผลลัพธ์ทั้ง 100 Generation ไปเขียนเป็นกราฟความสัมพันธ์ ดังภาพที่ 4.2



ภาพที่ 4.2 กราฟแสดงความเปลี่ยนแปลงใน 100 Generation

วิเคราะห์ผลการทดลอง

เมื่อทดลองให้โปรแกรมทำงานโดยเปลี่ยน Random Seed คำตอบที่เหมาะสมที่คำนวณได้ (Global optimum) ก็ยังวิ่งเข้าหาค่าที่ดีกว่าเมื่อ Generation เพิ่มขึ้น และค่า Global optimum จะคงที่เมื่อ การกระจายตัวของข้อมูลมีค่าน้อย นั่นคือประชากรที่มีอยู่ในขณะนั้นมีลักษณะเหมือนกันมาก จึงมีโอกาสที่จะพบค่าที่ดีกว่าได้น้อย

รอบที่ 3 ครั้งที่ 1 Number of Generation = 10

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 10

Minimize function Seed is : 90 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 46977

Traveling Route :

กรุงเทพ ลำปาง ชุมพร ชัยภูมิ ชัยนาท ตาก อ่างทอง สระแก้ว อำนาจเจริญ ภูเก็ต กาจนบุรี สมุทรปราการ กำแพงเพชร ตรัง จันทบุรี บุรีรัมย์ นครราชสีมา กาฬสินธุ์ น่าน ร้อยเอ็ด นครพนม ศรีสะเกษ นครนายก ลพบุรี หนองบัวลำภู เชียงใหม่ พิษณุโลก ลำพูน สิงห์บุรี อุดรธานี พิจิตร นครปฐม ประจวบคีรีขันธ์ ปทุมธานี มหาสารคาม เลยนนทบุรี อุบลราชธานี ชลบุรี กระบี่ ยโสธร สุรินทร์ ราชบุรี สุโขทัย สตูล พังงา อุทัยธานี เชียงราย สมุทรสงคราม ฉะเชิงเทรา อยุธยา มุกดาหาร อุตรดิตถ์ ปราจีนบุรี สุราษฎร์ธานี ระยอง สุพรรณบุรี หนองคาย สมุทรสาคร ระนอง พัทลุง นราธิวาส ปัตตานี แม่ฮ่องสอน นครสวรรค์ เพชรบุรี เพชรบูรณ์ พะเยา สระบุรี ยะลา แพร่ สกลนคร ขอนแก่น นครศรีธรรมราช ตราด สงขลา กรุงเทพ

รอบที่ 3 ครั้งที่ 2 Number of Generation = 25

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 25

Minimize function Seed is : 90 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 44183

Traveling Route :

กรุงเทพ เพชรบุรี ชัยภูมิ เชียงราย ชัยนาท ตราด อำนาจเจริญ นครสวรรค์ สระแก้ว ราชบุรี ลำพูน พะเยา กำแพงเพชร ระยอง เลย เชียงใหม่ นครราชสีมา ตรัง ร้อยเอ็ด อ่างทอง ภูเก็ต กาจนบุรี สมุทรปราการ ศรีสะเกษ ลำปาง สุโขทัย กาฬสินธุ์ พัทลุง สุราษฎร์ธานี บุรีรัมย์ อุดรธานี อุบลราชธานี พิจิตร ระนอง พังงา สมุทรสาคร ประจวบคีรีขันธ์ ลพบุรี อุทัยธานี มหาสารคาม หนองบัวลำภู นครพนม มุกดาหาร กระบี่ สตูล อุตรดิตถ์ สิงห์บุรี แพร่ สระบุรี นครนายก ฉะเชิงเทรา อยุธยา ชลบุรี ยโสธร นนทบุรี สุพรรณบุรี ปราจีนบุรี สุรินทร์ สกลนคร ชุมพร จันทบุรี เพชรบูรณ์ พิษณุโลก แม่ฮ่องสอน น่าน นครปฐม ตาก นครศรีธรรมราช ยะลา หนองคาย ขอนแก่น นราธิวาส สงขลา ปทุมธานี สมุทรสงคราม ปัตตานี กรุงเทพ

รอบที่ 3 ครั้งที่ 3 Number of Generation = 50

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 50

Minimize function Seed is : 90 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 43656

Traveling Route :

กรุงเทพ ชุมพร ลำปาง อุทัยธานี ราชบุรี นครสวรรค์ สระแก้ว ขอนแก่น ระยอง พะเยา เชียงราย น่าน กำแพงเพชร จันทบุรี สุพรรณบุรี ภูเก็ต นครราชสีมา ร้อยเอ็ด ศรีสะเกษ อ่างทอง อุดรธานี กาฬสินธุ์ พัทลุง ตรัง นราธิวาส อำนาจเจริญ ชลบุรี บุรีรัมย์ ปทุมธานี ชัยนาท

อุบลราชธานี สุโขทัย สมุทรปราการ ประจวบคีรีขันธ์ พิจิตร ลำพูน เชียงใหม่ สตูล พังงา ลพบุรี ตราด อุตรดิตถ์ นครพนม มหาสารคาม นครนายก หนองบัวลำภู มุกดาหาร แพร่ แม่ฮ่องสอน เพชรบูรณ์ กระบี่ ฉะเชิงเทรา ระนอง สมุทรสาคร นนทบุรี สุราษฎร์ธานี อยุธยา ปราจีนบุรี สิงห์บุรี เลย หนองคาย เพชรบุรี กาจนบุรี พิษณุโลก นครปฐม สุรินทร์ ยโสธร ตาก นครศรีธรรมราช ยะลา สงขลา สระบุรี สกลนคร ปัตตานี ชัยภูมิ สมุทรสงคราม กรุงเทพ

รอบที่ 3 ครั้งที่ 4 Number of Generation = 100

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 100

Minimize function Seed is : 90 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 43505

Traveling Route :

กรุงเทพ ชัยนาท พัทลุง ชุมพร ลำปาง อุทัยธานี อำนาจเจริญ ราชบุรี นครสวรรค์ สระแก้ว ขอนแก่น ระยอง พะเยา เชียงราย ตาก น่าน เชียงใหม่ กำแพงเพชร จันทบุรี ภูเก็ต พิษณุโลก ร้อยเอ็ด นครราชสีมา ศรีสะเกษ อ่างทอง กาฬสินธุ์ ตรัง ชลบุรี นราธิวาส บุรีรัมย์ ปทุมธานี อุบลราชธานี สุโขทัย สมุทรปราการ ประจวบคีรีขันธ์ ระนอง พิจิตร ลำพูน สตูล พังงา ตราด ลพบุรี นครพนม มหาสารคาม อุตรดิตถ์ นครนายก หนองบัวลำภู มุกดาหาร แพร่ เพชรบูรณ์ สมุทรสาคร กระบี่ สุราษฎร์ธานี ฉะเชิงเทรา ปราจีนบุรี นนทบุรี สุพรรณบุรี อยุธยา สิงห์บุรี เลย แม่ฮ่องสอน หนองคาย เพชรบุรี กาจนบุรี นครปฐม ยโสธร สุรินทร์ นครศรีธรรมราช ยะลา สงขลา สระบุรี อุดรธานี สกลนคร ชัยภูมิ สมุทรสงคราม ปัตตานี กรุงเทพ

ตารางที่ 4.3 ข้อมูลตั้งแต่ Generation ที่ 1 ถึง 100 (Seed 90)

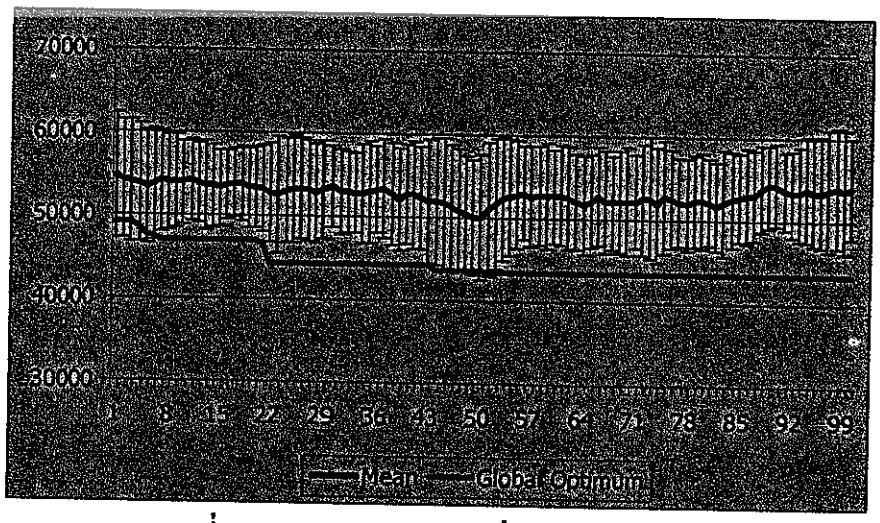
Generation	Local Optimum	Mean	SD	Global Optimum
1	49146	54782.97	2549.99	49146
2	49305	54339.04	2403.37	49146
3	49052	54101.97	2442.87	49052
4	48571	53901.17	2352.96	48571
5	47668	53418.92	2242.02	47668
6	47405	53747.47	2214.57	47405
7	46977	54224.67	2030.35	46977
8	48112	54032.17	1943.03	46977
9	48074	54144.64	1862.01	46977
10	47344	54041.31	1666.36	46977
11	49401	54181.78	1628.19	46977
12	49060	53908.7	1590.31	46977
13	47338	53552.06	1782.38	46977
14	49127	53528.92	1663.93	46977
15	48659	53352.69	1522.69	46977
16	49159	53697.95	1405.15	46977
17	48421	53888.14	1377.08	46977
18	49102	53849.26	1488.95	46977
19	47742	53360.74	1591.87	46977
20	47606	53347.95	1616.8	46977
21	46505	53238.34	1724.81	46505
22	44344	52714.59	1984.85	44344
23	44183	52589.59	2359.15	44183
24	44904	53065.72	2094.41	44183
25	45195	53244.44	2175.38	44183
26	45675	53266.62	2035.54	44183
27	46013	53092.24	2045.61	44183

Generation	Local Optimum	Mean	SD	Global Optimum
28	45350	52830.61	1989	44183
29	47385	53287.94	1983.82	44183
30	48460	53586.56	1654.67	44183
31	47272	52951.64	1699.41	44183
32	48400	52987.63	1646.42	44183
33	48040	52832.41	1602.24	44183
34	48202	52785.79	1983.54	44183
35	45470	52893.63	2053.65	44183
36	45951	53294.33	1761.16	44183
37	48783	53482.7	1980.64	44183
38	46714	52834.39	2054.66	44183
39	45245	52127.12	2087.32	44183
40	46062	52709.91	1995.4	44183
41	44940	52675.41	2135.72	44183
42	44218	52313.83	2083	44183
43	44196	51902.4	2486.23	44183
44	43656	51918.1	2556.28	43656
45	43959	51710.01	2596.09	43656
46	46032	51277.12	2482.56	43656
47	44827	50858.52	2514.49	43656
48	44756	50523.07	2333.43	43656
49	44560	50092.76	2335.66	43656
50	43903	50082.21	2499.73	43656
51	43505	51063.4	2697.21	43505
52	43597	51682.54	2602.11	43505
53	44553	52282.03	2477.37	43505
54	46286	52498.3	2334.12	43505
55	46894	52820.63	2155.02	43505

Generation	Local Optimum	Mean	SD	Global Optimum
56	47220	53016.39	1888.29	43505
57	45855	52645.47	2036.47	43505
58	45914	52807	1968.26	43505
59	46367	52699.21	2124.38	43505
60	45697	52669.7	1901.43	43505
61	47241	52817.38	2016.4	43505
62	47404	52521.68	2047.03	43505
63	47487	52120.1	1925.47	43505
64	45720	51672.42	1999.44	43505
65	47466	52188.79	1867.41	43505
66	47395	52727.1	2004.51	43505
67	46574	52129.95	2033.51	43505
68	47144	52048.25	1856.92	43505
69	46860	52138.34	2085.63	43505
70	45841	52229.98	2145.32	43505
71	45205	52200.95	1900.43	43505
72	46848	52505.06	2224.91	43505
73	46118	52615.22	2464.21	43505
74	45060	51847.56	2286.28	43505
75	45884	52677.24	2266.65	43505
76	46757	52623.36	1990.18	43505
77	47653	52094.35	1769	43505
78	47205	51938.85	1846.19	43505
79	46434	52341.02	1863.46	43505
80	47279	52282.05	1912.42	43505
81	47693	52173.45	1725.65	43505
82	46477	51677.91	1783.47	43505
83	45973	52092.02	2108.36	43505

Generation	Local Optimum	Mean	SD	Global Optimum
84	46543	52569.06	1886.17	43505
85	47113	52764.94	1802.67	43505
86	46012	52954.99	1906.89	43505
87	44272	53047.74	1820.85	43505
88	44236	53860.62	1747.51	43505
89	46074	54199.4	1670.97	43505
90	50798	54335.52	1679.22	43505
91	49725	53629.83	1564.27	43505
92	48809	53342.58	1720.84	43505
93	47270	53335.68	1857.71	43505
94	46664	53877.78	2121.55	43505
95	47915	53499.29	2149.42	43505
96	47357	53448.55	2231.23	43505
97	46933	53722.97	2444.11	43505
98	47271	54012.3	2425.57	43505
99	46259	53651.2	2500.24	43505
100	48219	53967.31	2021.29	43505

นำผลลัพธ์ทั้ง 100 Generation ไปเขียนเป็นกราฟความสัมพันธ์



ภาพที่ 4.3 กราฟแสดงความเปลี่ยนแปลงใน 100 Generation

วิเคราะห์ผลการทดลอง

จากกราฟนี้จะเห็นได้อย่างชัดเจนว่าช่วงที่ค่า Mean มีค่าต่ำสุดในช่วงที่การกระจายตัวสูงมากๆ เพราะว่าการกระจายที่สูงจะมีโอกาสที่จะทำให้ค่า Global Optimum ลดลงเนื่องจากประชากรมีความแตกต่างกันมาก

รอบที่ 4 ครั้งที่ 1 Number of Generation = 10

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 10

Minimize function Seed is : 12 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 45830

Traveling Route :

กรุงเทพ ระนอง นนทบุรี ขอนแก่น นครพนม อำนาจเจริญ ศรีสะเกษ อยุธยา ตราด สุโขทัย ตาก นราธิวาส เพชรบุรี ประจวบคีรีขันธ์ กาฬสินธุ์ ลพบุรี ชัยภูมิ อุบลราชธานี ยะลา สกลนคร สุพรรณบุรี พัทลุง เลย ลำพูน อุทัยธานี เพชรบูรณ์ สมุทรสงคราม สตูล กาญจนบุรี นครราชสีมา อ่างทอง อุตรดิตถ์ ปทุมธานี พิษณุโลก บุรีรัมย์ สระแก้ว จันทบุรี นครปฐม ชุมพร นครศรีธรรมราช สุรินทร์ สิงห์บุรี ปราจีนบุรี กระบี่ ชลบุรี ฉะเชิงเทรา สงขลา มหาสารคาม พิจิตร ราชบุรี นครสวรรค์ สุราษฎร์ธานี ภูเก็ต ตรัง ร้อยเอ็ด เชียงราย ชัยนาท สมุทรสาคร สมุทรปราการ ปัตตานี ลำปาง อุดรธานี หนองคาย ยโสธร หนองบัวลำภู มุกดาหาร เชียงใหม่ แม่ฮ่องสอน นครนายก สระบุรี น่าน พังงา พะเยา แพร่ กำแพงเพชร กรุงเทพ

รอบที่ 4 ครั้งที่ 2 Number of Generation = 25

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 25

Minimize function Seed is : 12 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 44039

Traveling Route :

กรุงเทพ มุกดาหาร บุรีรัมย์ อ่างทอง ปทุมธานี ราชบุรี สุโขทัย เพชรบูรณ์ แพร่ นครปฐม อุทัยธานี ปราจีนบุรี พิจิตร ฉะเชิงเทรา เชียงราย สงขลา พิษณุโลก เลย พัทลุง ชุมพร พะเยา นครนายก สิงห์บุรี ตราด นครสวรรค์ สมุทรสงคราม สระบุรี นครศรีธรรมราช พังงา ยะลา ระยอง ยโสธร สมุทรปราการ ภูเก็ต สุราษฎร์ธานี สตูล ประจวบคีรีขันธ์ หนองบัวลำภู กำแพงเพชร ปัตตานี กระบี่ ชลบุรี ลพบุรี ร้อยเอ็ด มหาสารคาม ลำพูน ลำปาง สุรินทร์ อุบลราชธานี นครพนม สกลนคร เชียงใหม่ อุดรธานี กาฬสินธุ์ ตาก จันทบุรี นราธิวาส เพชรบุรี อำนาจเจริญ หนองคาย สระแก้ว กาญจนบุรี ตรัง อุตรดิตถ์ ชัยภูมิ ขอนแก่น ชัยนาท น่าน สมุทรสาคร นนทบุรี แม่ฮ่องสอน นครราชสีมา ศรีสะเกษ ระนอง อยุธยา สุพรรณบุรี กรุงเทพ

รอบที่ 4 ครั้งที่ 3 Number of Generation = 50

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 50

Minimize function Seed is : 12 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 39729

Traveling Route :

กรุงเทพ พิจิตร ชัยนาท สระแก้ว ร้อยเอ็ด ประจวบคีรีขันธ์ ตราด อุทัยธานี ลำปาง ปราจีนบุรี ตรัง กาญจนบุรี อุบลราชธานี นครราชสีมา เชียงราย สระบุรี ชลบุรี นครนายก นครสวรรค์ หนองบัวลำภู เพชรบูรณ์ พิษณุโลก กาฬสินธุ์ เพชรบุรี อำนาจเจริญ ฉะเชิงเทรา อ่างทอง ภูเก็ต กระบี่ สุราษฎร์ธานี ปัตตานี สุพรรณบุรี จันทบุรี ลำพูน นครปฐม ยะลา ระยอง พะเยา นนทบุรี ตาก สมุทรสาคร พังงา

สมุทรสงคราม ปทุมธานี สิงห์บุรี นครพนม มุกดาหาร สตูล ยโสธร กำแพงเพชร ราชบุรี แม่ฮ่องสอน สุโขทัย หนองคาย แพร่ เชียงใหม่ อุดรธานี นราธิวาส พัทลุง นครศรีธรรมราช สงขลา ชุมพร เลย สกลนคร มหาสารคาม ชัยภูมิ อุตรดิตถ์ น่าน ลพบุรี บุรีรัมย์ ระนอง สุรินทร์ สมุทรปราการ อยุธยา ศรีสะเกษ ขอนแก่น กรุงเทพ

รอบที่ 4 ครั้งที่ 4 Number of Generation = 100

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 100

Minimize function Seed is : 12 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

Optimum Value :

Result = 43252

Traveling Route :

กรุงเทพ พิจิตร ชัยนาท สระแก้ว ร้อยเอ็ด ประจวบคีรีขันธ์ ตราด อุทัยธานี ลำปาง ปราจีนบุรี ตรัง กาจนบุรี อุบลราชธานี นครราชสีมา เชียงราย สระบุรี ชลบุรี นครนายก นครสวรรค์ หนองบัวลำภู เพชรบูรณ์ พิษณุโลก กาฬสินธุ์ เพชรบุรี อำนาจเจริญ ฉะเชิงเทรา อ่างทอง ภูเก็ต กระบี่ สุราษฎร์ธานี ปัตตานี สุพรรณบุรี จันทบุรี ลำพูน นครปฐม ยะลา ระยอง พะเยา นนทบุรี ตาก สมุทรสาคร พังงา สมุทรสงคราม ปทุมธานี สิงห์บุรี นครพนม มุกดาหาร สตูล ยโสธร กำแพงเพชร ราชบุรี แม่ฮ่องสอน สุโขทัย หนองคาย แพร่ เชียงใหม่ อุดรธานี นราธิวาส พัทลุง นครศรีธรรมราช สงขลา ชุมพร เลย สกลนคร มหาสารคาม ชัยภูมิ อุตรดิตถ์ น่าน ลพบุรี บุรีรัมย์ ระนอง สุรินทร์ สมุทรปราการ อยุธยา ศรีสะเกษ ขอนแก่น กรุงเทพ

ตารางที่ 4.4 ข้อมูลตั้งแต่ Generation ที่ 1 ถึง 100 (Seed 12)

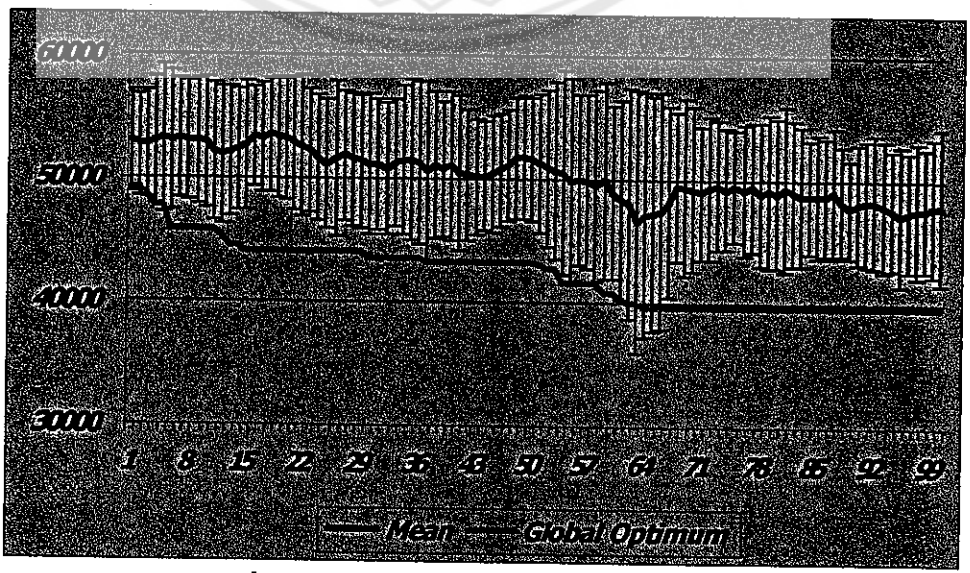
Generation	Local Optimum	Mean	SD	Global Optimum
1	49186	52988.51	1378.16	49186
2	49131	52670.39	1365.12	49131
3	48610	52806.92	1446.4	48610
4	47671	53040.9	1718.57	47671
5	47780	53173.84	1988.87	47671
6	45830	53023.4	1872.42	45830
7	48908	53171.54	1606.12	45830
8	47656	53007.32	1611.1	45830
9	47073	53020.63	1692.9	45830
10	46508	52834.99	1772.79	45830
11	46095	52176.81	1874.49	45830
12	45349	51858.24	1851.79	45349
13	44539	52158.88	1759.64	44539
14	44876	52345.84	1628.48	44539
15	44039	52707.62	1743.54	44039
16	47420	53338.91	1354.36	44039
17	47837	53259.12	1479.18	44039
18	47512	53522.64	1524.6	44039
19	45390	53356.36	1639.65	44039
20	44532	53157.08	1661.98	44039
21	47869	52851.11	1817.41	44039
22	46377	52653.77	1905	44039
23	46000	52230.69	1651.86	44039
24	47998	51685.77	1670.08	44039
25	46515	51167.2	1736.09	44039
26	46506	51612.28	2085.99	44039
27	46635	51861.15	1734.98	44039

Generation	Local Optimum	Mean	SD	Global Optimum
28	46176	51536.45	1788.44	44039
29	46123	51377.68	1790.21	44039
30	43521	51194.92	1906.1	43521
31	44768	51171.4	1775.89	43521
32	44152	50813.22	1765.8	43521
33	47109	51276.35	1736.43	43521
34	46786	51503.88	2028.49	43521
35	43778	51531.13	2166.73	43521
36	43545	51265.88	2210.36	43521
37	43252	50669.24	2419.43	43252
38	46178	51126.37	2014.63	43252
39	45632	50790.69	1820.65	43252
40	46242	51062.52	2033.26	43252
41	46143	50524.64	2107	43252
42	46432	50266.03	1777.39	43252
43	45681	50170.79	1569.71	43252
44	46718	50224.33	1491.76	43252
45	46438	50427.98	1578.2	43252
46	46091	50727.52	1516.69	43252
47	46291	51138.87	1492.62	43252
48	46300	51703.89	1660.87	43252
49	46401	51741.87	1671.72	43252
50	43791	51629.86	1680.88	43252
51	43118	51260.63	1886.37	43118
52	42785	50911.58	2125.11	42785
53	42690	50704.46	2390.18	42690
54	41693	50531.52	2784.2	41693
55	41908	49897.83	2787.3	41693

Generation	Local Optimum	Mean	SD	Global Optimum
56	42144	49995.89	2306.27	41693
57	41711	49853.87	2368.4	41693
58	41693	49462.94	2584.49	41693
59	40759	49945.06	2662.98	40759
60	40759	48951.51	2328.28	40759
61	40314	48374.08	2684.87	40314
62	40034	48074.27	3137.23	40034
63	40413	46581.01	3543.61	40034
64	39729	47162.42	3320.47	39729
65	39762	47245.44	3172.28	39729
66	40709	47491.89	3212.82	39729
67	40646	48250.53	2568.46	39729
68	43104	49452.33	2037.39	39729
69	42012	49424.06	2396.54	39729
70	44462	49257.46	2218.81	39729
71	43938	48984.83	1800.78	39729
72	43367	49338.73	1955.45	39729
73	43121	49585.65	1730.61	39729
74	43121	49269.08	1631.61	39729
75	44639	49513.36	1506.09	39729
76	42830	49195.84	1760.56	39729
77	42599	49453.17	1767.84	39729
78	42657	48884.13	1930.97	39729
79	43571	49114.93	2056.7	39729
80	45043	48924.03	2011.12	39729
81	44897	49277.51	2228.42	39729
82	44617	48981.67	1964.66	39729
83	44345	48683.21	1892.62	39729

Generation	Local Optimum	Mean	SD	Global Optimum
84	45365	48638.55	1518.74	39729
85	45261	48703.35	1676.17	39729
86	44282	48636.07	1569.6	39729
87	44991	48969.85	1737.52	39729
88	44711	48248.23	1473.22	39729
89	44663	47685.09	1324.41	39729
90	43114	48038.69	1606.53	39729
91	42697	48166.15	1678.57	39729
92	42450	48296.9	1802.32	39729
93	41021	47858.93	1782.62	39729
94	41482	47590.1	1611.49	39729
95	41058	47093.42	1884.01	39729
96	42227	47493.91	1612.59	39729
97	42010	47530.04	1824.38	39729
98	42441	47564.57	1687.93	39729
99	41521	47870.74	1844.14	39729
100	41195	47939.44	2089.19	39729

นำผลลัพธ์ทั้ง 100 Generation ไปเขียนเป็นกราฟความสัมพันธ์



ภาพที่ 4.4 กราฟแสดงความเปลี่ยนแปลงใน 100 Generation

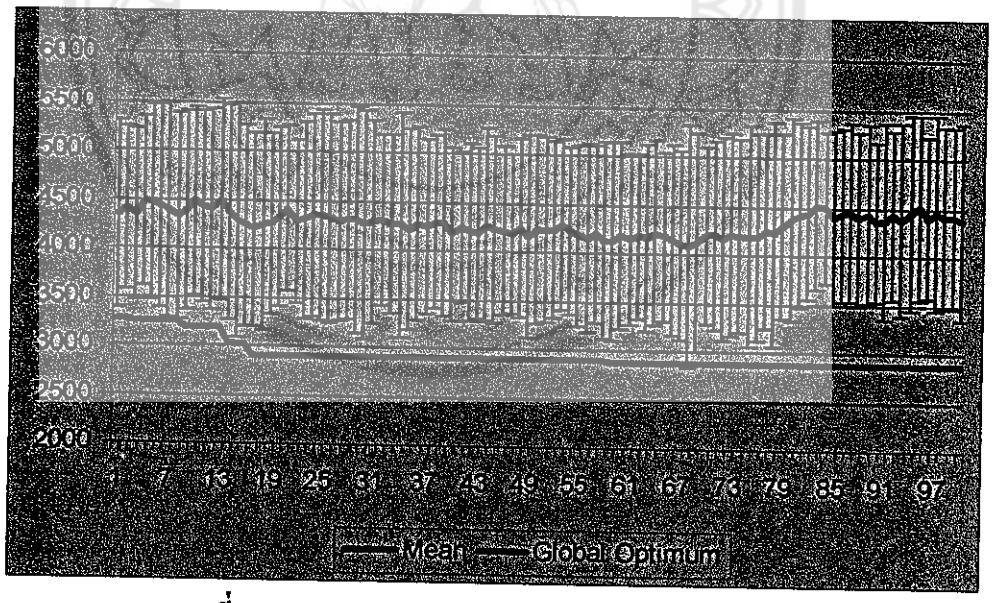
วิเคราะห์ผลการทดลอง

จากกราฟที่ได้นี้จะเห็นว่าเป็นกราฟที่ค่อนข้างจะดีมากเนื่องจากมีการกระจายตัวของข้อมูลมากในหลายๆ Generation ด้วยกันทำให้ค่า Global Optimum มีแนวโน้มที่ลดลงและค่าที่ต่ำที่สุดในนี้จะอยู่ในช่วงที่การกระจายมีค่ามากที่สุดด้วย จากการทดลองนี้จะพบตัวแปรที่สำคัญอีก 1 ตัว คือ จำนวนประชากร (เนื่องจากการกระจายตัวมากทำให้ประชากรหลากหลายมากขึ้น) ถ้าจำนวนประชากรมีค่ามากแล้วแสดงว่ามีโอกาสที่จะพบกับคำตอบที่ดีสูงด้วย

สรุปผลการทดลอง

จากการทดสอบทั้ง 4 รอบนั้นจะเห็นได้ว่ากราฟที่ได้มีความชันที่เป็นลบ นั่นคือคำตอบที่ได้พยายามวิ่งเข้าหาค่าต่ำสุด(เนื่องจากทดสอบด้วยฟังก์ชัน Minimize) จึงแสดงให้เห็นว่าโปรแกรมที่สร้างขึ้นมานั้นมีศักยภาพในการหาคำตอบที่เหมาะสมได้

และเพื่อให้ง่ายต่อความเข้าใจเราจึงนำข้อมูลมาเพียงเฉพาะภาคเหนือและ Run โปรแกรมทดสอบหาค่าการเดินทางที่ต่ำสุด และนำคำตอบที่ได้มาเขียนเส้นทางการเดินทางซึ่งแสดงผลออกมาดังต่อไปนี้



ภาพที่ 4.5 กราฟของระยะทางการเดินทางเฉพาะภาคเหนือ

ที่ Generation ที่ 1 นำค่าต่ำสุดมาเขียนเส้นทางเดินได้ดังภาพที่ 4.6

ภาพที่ 4.6 แผนภาพแสดงเส้นทางเดินภายในภาคเหนือจากคำตอบที่ 1st Generation

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 15

Number of Generation : 1

Minimize function Seed is : 95 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at 2

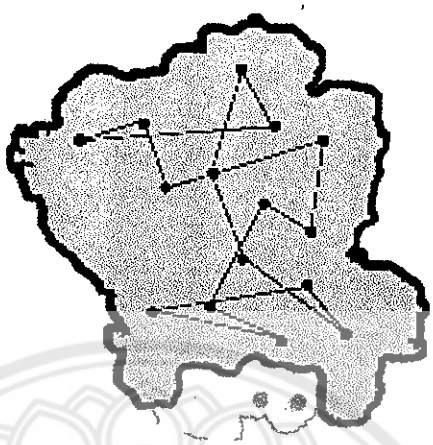
Optimum Value :

Result = 3267

Traveling Route :

เชียงราย พะเยา พิษณุโลก ลำปาง ลำพูน เชียงใหม่ แม่ฮ่องสอน น่าน เพชรบูรณ์ พิจิตร ตาก สุโขทัย กำแพงเพชร แพร่ อุตรดิตถ์ เชียงราย

ที่ Generation ที่ 14 นำค่าต่ำสุดมาเขียนเส้นทางเดินได้ดังภาพที่



ภาพที่ 4.7 แผนภาพแสดงเส้นทางเดินภายในภาคเหนือจากคำตอบที่ 14th Generation

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 15

Number of Generation : 14

Minimize function Seed is : 95 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

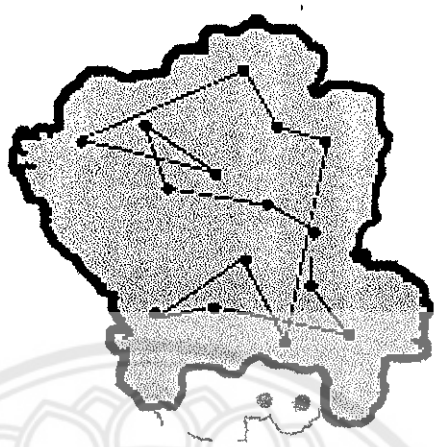
Optimum Value :

Result = 3026

Traveling Route :

เชียงราย พะเยา แม่ฮ่องสอน เชียงใหม่ ลำพูน น่าน อุตรดิตถ์ แพร่ กำแพงเพชร พิจิตร ตาก พิษณุโลก เพชรบูรณ์ สุโขทัย ลำปาง เชียงราย

ที่ Generation ที่ 59 นำค่าต่ำสุดมาเขียนเส้นทางเดินได้ดังภาพที่ 4.8



ภาพที่ 4.8 แผนภาพแสดงเส้นทางเดินภายในภาคเหนือจากคำตอบที่ 59th Generation

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 15

Number of Generation : 59

Minimize function Seed is : 95 Selection based on : Regular Sampling space

Close Loop Started at : 2 Finished at : 2

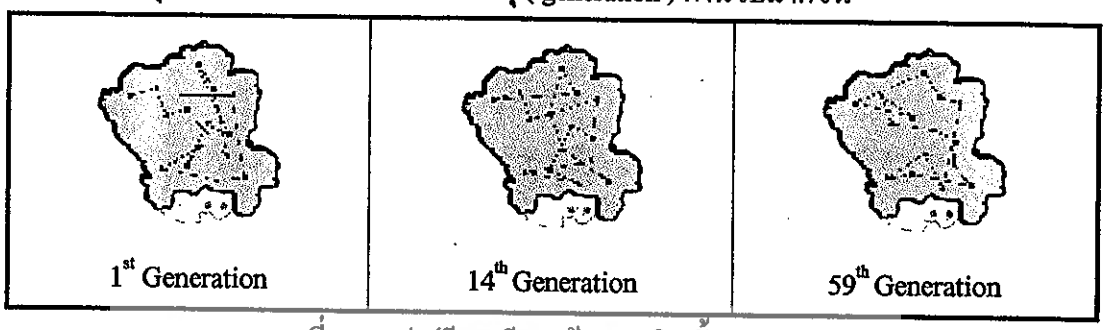
Optimum Value :

Result = 2875

Traveling Route :

เชียงราย แม่ฮ่องสอน ลำปาง เชียงใหม่ ลำพูน แพร่ อุตรดิตถ์ พิษณุโลก เพชรบูรณ์ กำแพงเพชร ตาก สุโขทัย พิจิตร น่าน พะเยา เชียงราย

จากภาพที่(4.9) ทางด้านล่างนี้จะเห็นได้ว่าเส้นทางเดินจะเริ่มจากมีทางเดินที่สับสนและจะค่อยๆปรับปรุงให้เป็นระเบียบขึ้นเมื่อช่วงอายุ (generation) ผ่านไปมากขึ้น



ภาพที่ 4.9 รูปเปรียบเทียบเส้นทางเดินทั้ง 3 Generation

จึงเป็นข้อสรุปที่เห็นได้ชัดว่าโปรมแกรมสามารถนำไปสู่คำตอบที่ดีขึ้นได้เมื่อผ่านไปหลายๆ Generation นั่นคือถ้าเรากำหนดค่า Number of generation ที่มากๆ ก็ย่อมจะมีโอกาสพบกับคำตอบที่ดีมากขึ้นด้วย

เปรียบเทียบการใช้ความคิดวิเคราะห์เส้นทาง

เชียงราย เชียงใหม่ แม่ฮ่องสอน ลำพูน ลำปาง พะเยา น่าน แพร่ อุตรดิตถ์ พิษณุโลก เพชรบูรณ์ พิจิตร กำแพงเพชร ตาก สุโขทัย เชียงราย

ระยะทางรวม 2447

จากการคำนวณด้วยความคิดนี้ผลลัพธ์ที่ได้กลับดีกว่าที่คำนวณได้จากโปรแกรม เนื่องจากผู้ที่ทำการทดสอบด้วยความคิดนี้มีประสบการณ์ในการเดินทางในภาคเหนือบ้างประกอบกับโปรแกรมทำการสุ่มข้อมูลมาทั้งหมด 100 population x 59 generation = 5900 population จากจำนวนทั้งหมดที่เป็นไปได้ 3,220,801,968,384,000 ชุด ซึ่งถือว่าคำตอบที่ได้ใกล้เคียงกับผู้มีประสบการณ์พอควร และถ้าสามารถกำหนดค่า population size ให้มากขึ้นย่อมมีโอกาสจะพบคำตอบที่ดีขึ้นกว่านี้

4.3 การทดลองเพื่อหาการกำหนดค่าที่เหมาะสมภายใต้ข้อจำกัด

จากการทฤษฎีโดยทั่วไปของ Genetic Algorithm และผลการคำนวณจากการทดลองที่ผ่านมาสามารถบอกได้ว่า เมื่อ Generation สูงขึ้นจะมีโอกาสพบคำตอบที่ดีขึ้น แต่การคำนวณที่จำนวน Generation สูงๆนั้นจะต้องใช้เวลาในการคำนวณสูงตามไปด้วย ดังนั้นถ้าหากกรณีที่ถูกจำกัดด้วยเวลาแล้วจึงไม่สามารถจะกำหนดค่า Number of Generation ได้ตามความต้องการอย่างแน่นอน ดังนั้นจึงได้นำค่าการกำหนดจากการทดลองที่แล้วมาทดลองซึ่งมีการกำหนดการตั้งค่าดังนี้

- Genetic operation method

>Position-based Crossover (P.Pongcharoen 2001)

>Two Operation Adjacent Swap Mutation (P.Pongcharoen 2001)

- Selection based on regular sampling space (P.Pongcharoen 2001)
- 90% Crossover rate, 18% Mutation rate (P.Pongcharoen 2001)
- Number of gene = 76
- Minimize function
- Close loop, Started and finished at 2(กรุงเทพฯ)
- Specific random Seed = 2, 12, 69, 90

ตัวแปรที่กำหนดเป็นชุดทดสอบคือ Population size และ Number of Generation โดยกำหนดค่าดังนี้

P/G = 100/50 และ P/G = 50/100 ซึ่งเมื่อรวมและจะมีกลุ่มประชากรทั้งหมดเท่ากันคือ 5,000 ชุด ซึ่งทำให้เป็นการเปรียบเทียบที่เท่าเทียมกัน

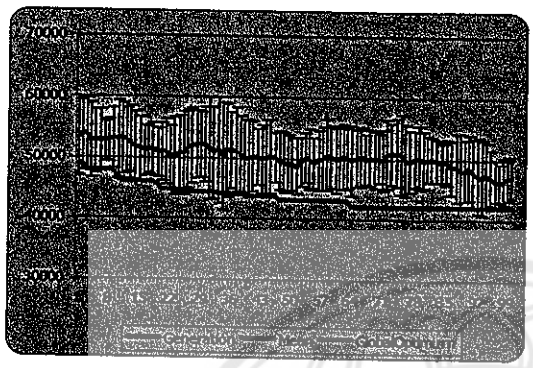
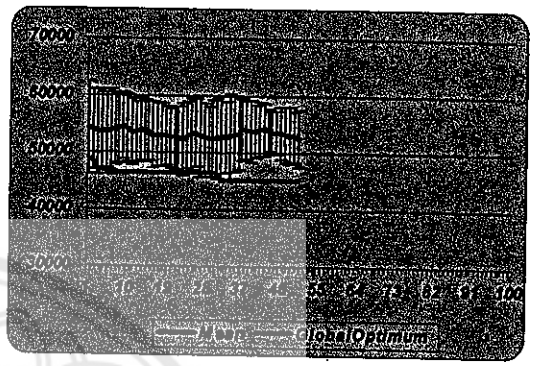
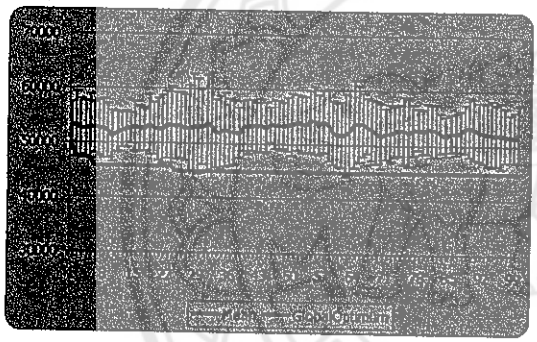
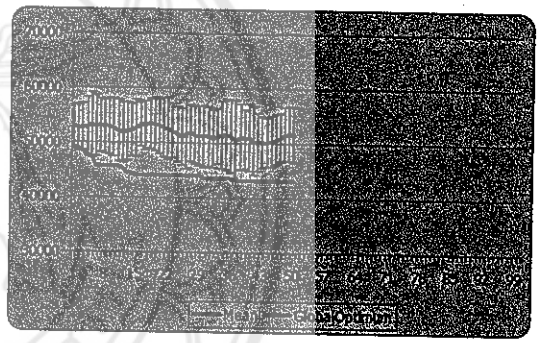
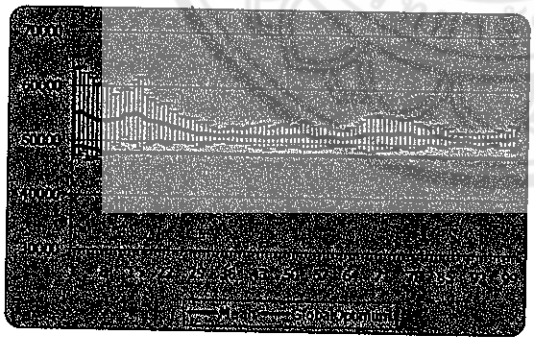
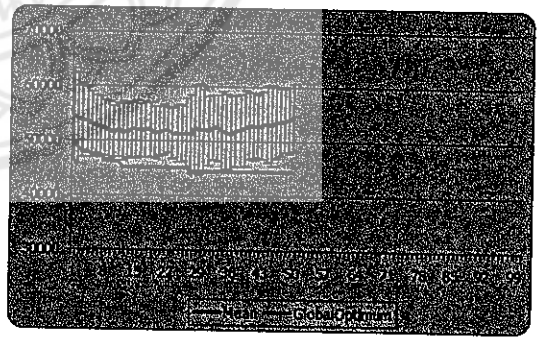
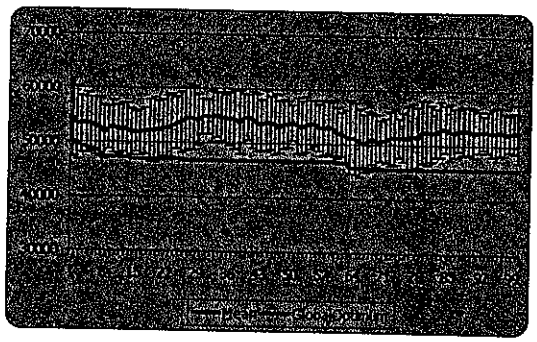
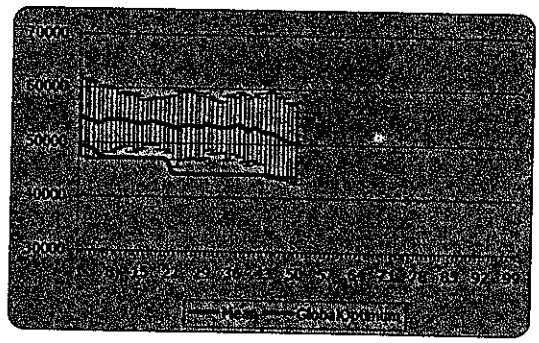
ตารางที่ 4.5 ข้อมูลของเครื่อง Computer ที่ใช้ในการทดสอบ

รายการ	รายละเอียด
CPU	AMD Athlon XP 1800+ (1.533 GHz)
Hard disk	60 Gb ATA 100
Ram	512 MB
ระบบปฏิบัติการ	Microsoft Window XP Professional
ระบบจัดการฐานข้อมูล	Microsoft office XP

ตารางที่ 4.6 ผลการทดลอง

Seed	หัวข้อเปรียบเทียบ	P/G	
		100/50	50/100
2	เวลาที่ใช้ (นาที)	3:09	3:13
	คำตอบที่ได้ (km)	44743	42131
12	เวลาที่ใช้ (นาที)	3 :16	3 :23
	คำตอบที่ได้ (km)	43252	44864
69	เวลาที่ใช้ (นาที)	3 :11	3 :12
	คำตอบที่ได้ (km)	44529	47779
90	เวลาที่ใช้ (นาที)	3 :11	3 :11
	คำตอบที่ได้ (km)	43656	45588

ตารางที่ 4.7 ภาพชุดเปรียบเทียบผลลัพธ์จากการทดลองการกำหนดค่าที่เหมาะสมภายใต้ข้อจำกัด

P/G 50/100	P/G 100/50
Seed 2	Seed 2
	
Seed 12	Seed 12
	
Seed 69	Seed 69
	
Seed 90	Seed 90
	

วิเคราะห์ผลการทดลอง

การทดลองนี้ถูกกระทำบนเครื่องคอมพิวเตอร์ที่มีประสิทธิภาพค่อนข้างสูงจึงทำให้ระยะเวลาการคำนวณต่างกันไม่มากนัก เรียกได้ว่าต่างกันเพียงตัวเลขแต่ไม่แตกต่างกันทางความรู้สึก แต่ผลของคำตอบที่ได้มานั้นมีความแตกต่างกันอยู่พอสมควรเลยที่เดียว และเนื่องจาก Genetic Algorithm นี้เป็นกระบวนการแก้ปัญหาแบบไม่แน่นอนไม่สามารถทำให้ตายตัวได้ ดังนั้นคำตอบที่ได้อาจจะมากกว่ากันบ้างหรือน้อยกว่ากันบ้างจึงไม่ใช่เรื่องแปลกหรือผิดแต่อย่างใด และเมื่อนำค่าในทุกๆ Generation มาเขียนกราฟเปรียบเทียบด้วยมาตราส่วนที่เท่ากันแล้วพบว่า กราฟของตัวแปร P/G = 100/50 จะมีความชันเป็นลบมากกว่า P/G = 50/100 นั่นแสดงถึงว่าการคำนวณจะสามารถเข้าหาคำตอบที่ดีได้เร็วกว่าด้วยเช่นกัน(ในกรณีที่หาค่าน้อยที่สุด) และเมื่อสังเกตการณ์กระจายตัวของข้อมูลจะพบอีกว่า การกระจายตัวของข้อมูลของ P/G = 100/50 จะมีมากกว่าทำให้มีโอกาสที่จะได้ Chromosome ที่แข็งแรงมีมากขึ้น และเมื่อการกระจายตัวของข้อมูลต่ำแสดงว่า Chromosome แต่ละตัวมีรูปร่างหน้าตาคล้ายๆกันเมื่อผ่านกระบวนการตัดสลับ(Crossover)แล้วถูกที่ได้จะมีการเปลี่ยนแปลงไม่มากทำให้ความแข็งแรงของChromosome มีค่าเท่าๆกับของเดิม ดังนั้นจึงต้องพึ่งกระบวนการกลายพันธุ์เท่านั้นที่จะได้ค่าที่แตกต่างจากรุ่นพ่อแม่อย่างชัดเจน ดังนั้นการตั้งค่าให้จำนวนประชากรสูงๆจะทำให้มีโอกาสที่จะเจอคำตอบที่เหมาะสมเร็วกว่า

สรุปผลการทดลอง

การทดลองนี้ทำให้เห็นว่าถ้าหากมีข้อจำกัดด้านเวลาควรที่จะตั้งค่าให้มีจำนวนประชากรสูงๆ ซึ่งจะให้ผลลัพธ์ที่ดีกว่าและใช้เวลาที่น้อยกว่า ในการทดลองนี้อาจจะไม่เห็นความแตกต่างทางด้านเวลาเพราะเครื่องคอมพิวเตอร์ที่ใช้ทดสอบมีประสิทธิภาพสูง แต่หากนำไปทำงานในเครื่องคอมพิวเตอร์ที่มีทรัพยากรของเครื่องน้อย ตัวแปรทางด้านเวลานี้จะเพิ่มขึ้นอีกจนเป็นส่วนสำคัญในการพิจารณาเลยทีเดียว ดังข้อมูลอ้างอิงนี้

ตารางที่ 4.8 Spec เครื่องที่ใช้เทียบอ้างอิงเวลา

รายการ	รายละเอียด
CPU	Intel Pentium III 450 MHz
Hard disk	40 Gb ATA 33
Ram	196 MB
ระบบปฏิบัติการ	Microsoft Window ME
ระบบจัดการฐานข้อมูล	Microsoft office XP

P/G = 100/50 ใช้เวลาคำนวณเฉลี่ย 23:56 นาที , P/G = 50/100 ใช้เวลาคำนวณเฉลี่ย 26:39 นาที

4.4 การทดลองเกี่ยวกับฐานของการคัดสรร Selection Base

เนื่องจากในการทดลองที่ 4.2 นั้นแสดงให้เห็นว่าถ้าจำนวนประชากรในการคัดเลือกต่อรอบมีจำนวนมากจะทำให้มีโอกาสที่จะพบคำตอบที่ดีมากขึ้นด้วย และเนื่องจากโปรแกรมนี้มีฐานการคัดเลือกอยู่ทั้งหมด 2 แบบด้วยกัน คือ แบบลูกทดแทนพ่อแม่(Regular Sampling Space) และ แบบลูกรวมกับพ่อแม่ (Enlarge Sampling Space) ในการทดลองที่ผ่านมาจะใช้ Regular Sampling Space เนื่องจากในงานวิจัย(P.Pongcharoen 2001) ที่ได้อ้างอิงการกำหนดตัวแปรเริ่มต้นมาใช้นั้น ฟังก์ชันการเลือกสรรของบทความนั้นอยู่บนฐานของลูกทดแทนพ่อแม่ ดังนั้นในการทดลองครั้งนี้จึงนำเอาการทดลองที่ 4.1 มาทดลองอีก 1 ครั้งโดยใช้ฐานการคัดสรรแบบ Enlarge Sampling Space และกำหนดช่วงอายุที่ 100 Generation ซึ่งได้ผลการทดลองดังนี้

ผลการทดลอง

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 100

Minimize function Seed is : 2 Selection based on : Enlarged Sampling space

Close Loop Started at : 2 Finished at : 2

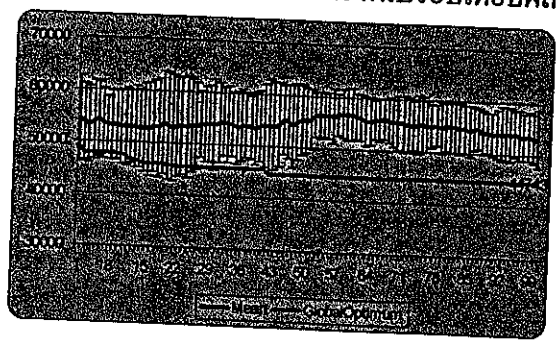
Optimum Value :

Result = 44743

Traveling Route :

กรุงเทพ อ่างทอง พังงา ชุมพร กาจนบุรี นครปฐม ศรีสะเกษ สระบุรี ปราจีนบุรี น่าน ลำปาง เลย ลำพูน อุทัยธานี สระแก้ว จันทบุรี มุกดาหาร ตาก บุรีรัมย์ ระนอง สุราษฎร์ธานี พิษณุโลก อุบลราชธานี เชียงใหม่ อุตรดิตถ์ ฉะเชิงเทรา ภูเก็ต ยะลา นครนายก หนองบัวลำภู สมุทรสงคราม ปทุมธานี สุพรรณบุรี อำนาจเจริญ ชัยนาท นราธิวาส นครสวรรค์ ลพบุรี นครราชสีมา ราชบุรี ระยอง นนทบุรี นครศรีธรรมราช ประจวบคีรีขันธ์ ยโสธร สกลนคร อุดรธานี กาฬสินธุ์ เชียงราย นครพนม หนองคาย สงขลา สิงห์บุรี เพชรบุรี สมุทรสาคร พะเยา ปัตตานี ตรัง สตูล ตราด ร้อยเอ็ด พิจิตร สุโขทัย ขอนแก่น พัทลุง กำแพงเพชร แม่ฮ่องสอน สมุทรปราการ กระบี่ ชลบุรี แพร่ ชัยภูมิ มหาสารคาม เพชรบูรณ์ อยุธยา สุรินทร์ กรุงเทพ

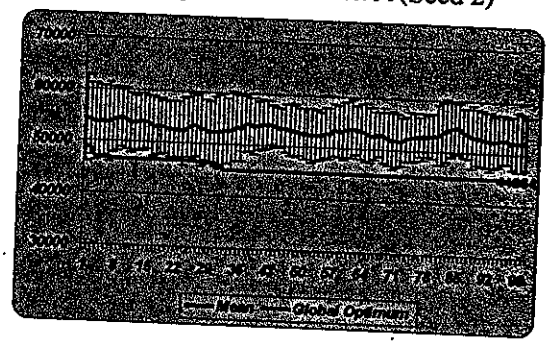
ภาพที่ 4.10 กราฟเปรียบเทียบผลลัพธ์การกำหนดฐานของการคัดสรร(Seed 2)



Enlarge Sampling Space

Minimum = 44743

Time = 6.29



Regular Sampling Space

Minimum = 44664

Time = 6.26

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 100

Minimize function Seed is : 12 Selection based on : Enlarged Sampling space

Close Loop Started at : 2 Finished at : 2

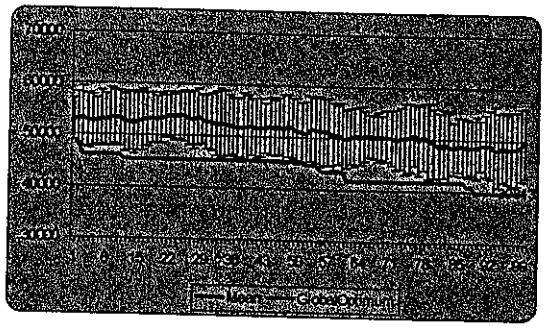
Optimum Value :

Result = 39813

Traveling Route :

กรุงเทพ ราชบุรี สระแก้ว ชัยนาท ปทุมธานี ฉะเชิงเทรา ตราด หนองบัวลำภู ชลบุรี นครพนม ตรัง กระบี่ กาจนบุรี ปราจีนบุรี เพชรบุรี นครสวรรค์ ยโสธร บุรีรัมย์ เพชรบูรณ์ สุพรรณบุรี ลำปาง พะเยา นครปฐม จันทบุรี ลำพูน กาฬสินธุ์ สุราษฎร์ธานี นราธิวาส อ่างทอง เชียงราย อำนาจเจริญ ระยอง นครนายก สมุทรสงคราม สมุทรสาคร สมุทรปราการ ยะลา ปัตตานี พังงา พิจิตร ประจวบคีรีขันธ์ ตาก น่าน ศรีสะเกษ ภูเก็ต ระนอง มุกดาหาร อุบลราชธานี ร้อยเอ็ด สุรินทร์ อุดรธานี หนองคาย เชียงใหม่ อุทัยธานี สุโขทัย สตูล พัทลุง พิษณุโลก สกลนคร ขอนแก่น มหาสารคาม เลย ชุมพร สระบุรี แม่ฮ่องสอน กำแพงเพชร แพร่ สิงห์บุรี อุตรดิตถ์ ชัยภูมิ ลพบุรี อยุธยา นครราชสีมา นนทบุรี นครศรีธรรมราช สงขลา กรุงเทพ

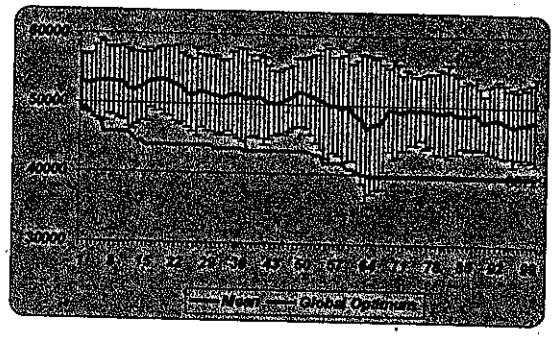
ภาพที่ 4.11 กราฟเปรียบเทียบผลลัพธ์การกำหนดฐานของการคัดสรร(Seed 12)



Enlarge Sampling Space

Minimum = 39813

Time = 6.28



Regular Sampling Space

Minimum = 39729

Time = 6.27

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 100

Minimize function Seed is : 69 Selection based on : Enlarged Sampling space

Close Loop Started at : 2 Finished at : 2

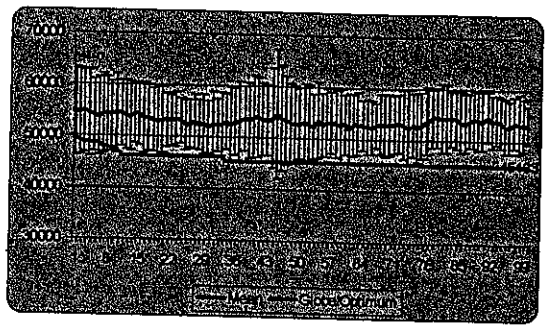
Optimum Value :

Result = 45133

Traveling Route :

กรุงเทพ ตาก อุดรธานี อำนาจเจริญ ระยอง อุตรดิตถ์ กาจนบุรี เพชรบูรณ์ สุโขทัย สระแก้ว ปทุมธานี กระบี่ เชียงราย ร้อยเอ็ด อยุธยา สระบุรี ยโสธร หนองคาย เชียงใหม่ พิจิตร น่าน พิษณุโลก เลย อุบลราชธานี มหาสารคาม พังงา นครราชสีมา ประจวบคีรีขันธ์ สตูล นราธิวาส กาฬสินธุ์ ยะลา ตรัง หนองบัวลำภู กำแพงเพชร สมุทรปราการ แพร่ ลพบุรี นครปฐม ปราจีนบุรี มุกดาหาร บุรีรัมย์ อ่างทอง นครศรีธรรมราช ชุมพร ชัยนาท ศรีสะเกษ นนทบุรี อุทัยธานี สงขลา พัทลุง ฉะเชิงเทรา ตราด ขอนแก่น สมุทรสงคราม เพชรบุรี สุรินทร์ นครพนม สมุทรสาคร พะเยา แม่ฮ่องสอน สกลนคร สุพรรณบุรี ระนอง ปัตตานี ลำปาง ชลบุรี ภูเก็ต สุราษฎร์ธานี ราชบุรี สิงห์บุรี จันทบุรี นครนายก ชัยภูมิ ลำพูน นครสวรรค์ กรุงเทพ

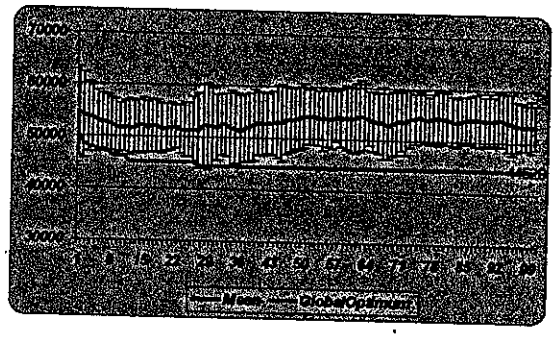
ภาพที่ 4.12 กราฟเปรียบเทียบผลลัพธ์การกำหนดฐานของการคัดสรร(Seed 69)



Enlarge Sampling Space

Minimum = 45133

Time = 6.26



Regular Sampling Space

Minimum = 44529

Time = 6.26

Data Report

Genetic Parameter :

Population Size : 100 / 90% Position-base Crossover /18% 2-Operation Adjacent Swap Mutation

Number of Gene : 76

Number of Generation : 100

Minimize function Seed is : 90 Selection based on : Enlarged Sampling space

Close Loop Started at : 2 Finished at : 2

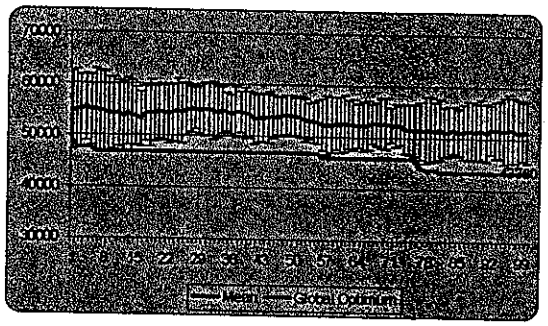
Optimum Value :

Result = 43360

Traveling Route :

กรุงเทพ แพร่ พิจิตร สิงห์บุรี บุรีรัมย์ อุบลราชธานี ชัยนาท ชัยภูมิ ปทุมธานี นราธิวาส ยะลา ลำปาง สมุทรสาคร มหาสารคาม นครราชสีมา อุดรธานี ภูเก็ต สระบุรี สมุทรปราการ นนทบุรี ตรัง สุพรรณบุรี กระบี่ อุตรดิตถ์ อุทัยธานี ลำพูน นครศรีธรรมราช สงขลา ปัตตานี อำนาจเจริญ ประจวบคีรีขันธ์ นครนายก พัทลุง นครสวรรค์ อยุธยา สุโขทัย ระนอง ลพบุรี สระแก้ว ชลบุรี สตูล พังงา สุรินทร์ ศรีสะเกษ ยโสธร หนองบัวลำภู ราชบุรี สุราษฎร์ธานี สมุทรสงคราม น่าน นครปฐม ร้อยเอ็ด มุกดาหาร ขอนแก่น ระยอง หนองคาย ฉะเชิงเทรา จันทบุรี ตราด อ่างทอง พิษณุโลก ตาก พะเยา เพชรบูรณ์ ชุมพร เพชรบุรี นครพนม กาฬสินธุ์ กำแพงเพชร แม่ฮ่องสอน เชียงใหม่ เชียงราย สกลนคร ปราจีนบุรี กาญจนบุรี เลย กรุงเทพ

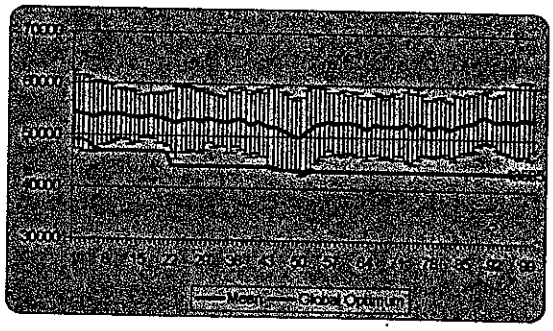
รูปที่ 4.13 กราฟเปรียบเทียบผลลัพธ์การกำหนดฐานของการคัดสรร(Seed 90)



Enlarge Sampling Space

Minimum = 43360

Time = 6.26



Regular Sampling Space

Minimum = 43505

Time = 6.24

วิเคราะห์ผลการทดลอง

จากผลการทดลองที่ปรากฏเป็นที่น่าสังเกตว่าคำตอบที่ได้จากการคัดสรรบนฐานของ Enlarge Sampling Space จะได้คำตอบที่ไม่ดีนักเมื่อเทียบกับคำตอบของการคัดสรรบนฐานของ Regular Sampling Space ซึ่งเป็นสาเหตุมาจากการฟังก์ชันการคัดสรรที่โปรแกรมใช้อยู่นี้เป็นแบบ Roulette Wheel ซึ่งให้ทุกๆ โครมโมโซมจะมีโอกาสได้ผ่านไปยังช่วงอายุต่อไป (Next Generation) แต่โอกาสจะมากจะน้อยนั้นขึ้นอยู่กับความแข็งแรงของ Chromosome จึงเป็นไปได้ที่ Seed ที่ใช้อยู่ในการทดสอบนี้จะคัดเลือกเอา Chromosome ไม่ค่อยแข็งแรงไปมากจึงทำให้ลูกในแต่ละช่วงอายุพัฒนาได้ช้าจึงเป็นต้นเหตุให้ผลลัพธ์วิ่งเข้าสู่ค่าที่ดีได้ช้า แต่ค่ากลางของในแต่ละช่วงอายุจะค่อนข้างเป็นเส้นตรงกว่าแบบ Regular Sampling Space เนื่องจากการที่ Chromosome ผ่านกระบวนการตัดสลับ (Crossover) หรือกระบวนการกลายพันธุ์ (Mutation) ไม่ได้หมายความว่าลูกที่ออกมาจะต้องมีค่าที่ดีขึ้นเสมอไป ดังนั้นแบบ Regular Sampling Space จึงการเปลี่ยนแปลงของค่ากลางมากกว่าแบบ Enlarge Sampling Space ในส่วนของเวลาที่ใช้คำนวณก็ใกล้เคียงกันมากจนแทบไม่มีผลในสำหรับคอมพิวเตอร์ที่มีทรัพยากรสูง

สรุปผลการทดลอง

การคัดสรรบนฐานแบบลูกรวมกับพ่อแม่ (Enlarge Sampling Space) จะได้ผลดีมากกว่าเมื่อใช้กับฟังก์ชันการคัดสรรแบบคัดเลือกตัวที่แข็งแรงเท่านั้นที่จะผ่านไปได้ แต่เมื่อนำมาใช้กับ Roulette Wheel จึงยังไม่อาจจะเห็นผลได้ชัดเจน และจากการทดลองพบว่าแบบ Enlarge Sampling Space นั้นจะใช้ระยะเวลามากกว่าเล็กน้อยดังนั้นหากทำบนเครื่องคอมพิวเตอร์ที่มีทรัพยากรน้อยก็ไม่ควรใช้ตัวเลือก Enlarge Sampling Space

4.5 สรุปการทำงานทั้งหมดของโปรแกรม

คำตอบที่ได้จากการคำนวณนี้อาจเป็นเพียงตัวเลือกหนึ่งในหลายๆตัวเลือกในการตัดสินใจเท่านั้นเพราะบางทีอาจมีความสัมพันธ์อย่างอื่นที่ควรนำมาวิเคราะห์ในการตัดสินใจด้วย และเนื่องจาก Genetic Algorithm นี้เป็นกระบวนการหาคำตอบแบบไม่แน่นอน (Stochastic search) จึงทำให้ในแต่ละครั้งอาจจะได้คำตอบที่ดีกว่าเดิมก็ได้ซึ่งอยู่ที่ลำดับการสุ่มของ Seed และ Genetic algorithm นี้ไม่สามารถการันตรีได้ว่าคำตอบที่ได้นั้นเป็นคำตอบที่ดีที่สุดแล้ว วิธีเดียวที่จะสามารถหาคำตอบที่ดีที่สุดได้คือ การหาคำตอบทุกๆตัวที่เป็นไปได้ของสมการซึ่งในกรณีศึกษาการเดินทางภายในประเทศไทยนี้จะต้องจะต้องหาคำตอบทั้งหมด 76! ซึ่งจำนวนของคำตอบนั้นมากมายมหาสารประมาณ 1.885×10^{111} คำตอบซึ่งคงต้องใช้เวลานานมากในการหาคำตอบทั้งหมดดังนั้น Genetic Algorithm นี้เป็นทางเลือกหนึ่งที่จะสามารถหาคำตอบที่เหมาะสมได้ในขอบเขตที่มีทรัพยากรต่างๆจำกัดโดยเฉพาะเวลา ซึ่งสามารถได้คำตอบที่จะยอมรับได้ในระยะเวลาอันสั้น และโปรแกรมชุดนี้ก็พิสูจน์ให้เห็นแล้วว่ามีศักยภาพในการหาคำตอบที่เหมาะสมตามจุดประสงค์ได้

บทที่ 5
สรุปผลการดำเนินงานวิจัยและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงานวิจัย

5.1.1 โปรแกรมที่สร้างมีศักยภาพในการหาคำตอบที่เหมาะสม

5.1.2 การทำงานสามารถใช้งานได้ง่ายมีฟังก์ชันอำนวยความสะดวก เช่นพิมพ์ผลลัพธ์ได้ทันที บันทึกข้อมูลที่คำนวณได้

5.1.3 วิเคราะห์หาวิธีการใช้งานที่ได้คำตอบที่เหมาะสมและประหยัดเวลา คือการตั้งค่า Number of population มากๆ จะได้ผลดีกว่าตังให้ Number of Generation มากๆ

5.1.4 ทดสอบการทำงานของฐานการคัดสรรแบบ Enlarge sampling space เมื่อทำงานคู่กับฟังก์ชันการคัดสรรแบบ Roulette Wheel

5.1.5 การนำโปรแกรมนี้ไปใช้นั้นไม่ได้ใช้ได้เฉพาะกรณีศึกษานี้เท่านั้นโปรแกรมสามารถเปลี่ยนฐานข้อมูลในการคำนวณได้โดยผู้ใช้งาน ตัวอย่างเช่น

1) สามารถนำไปประยุกต์ใช้ในโรงงานอุตสาหกรรมคอมพิวเตอร์ คือในการเชื่อม Print Circuit Board นั้นจะมีจุดต่างๆที่ต้องการเชื่อมมากมายดังนั้นถ้าคำนวณระยะทางรวมที่สั้นที่สุดในการเชื่อมย่อมทำให้ลดเวลาลดค่าใช้จ่ายลงได้

2) ประยุกต์ใช้กับตารางความสัมพันธ์แบบ Form to Chart ได้

5.2 ข้อแนะนำ

5.2.1 ในโปรแกรมชุดนี้จะใช้ฟังก์ชันการคัดสรรแบบ Roulette Wheel ซึ่งเป็นกระบวนการที่เลียนแบบธรรมชาติ ถ้าเราทำฟังก์ชันนี้ให้มีบังคับการคัดสรรเฉพาะตัวที่แข็งแรงจึงมีความเป็นไปได้ที่จะได้คำตอบที่ดีขึ้นมากกว่าเดิม และจุดสำคัญอีกหนึ่งอย่างก็คือในกระบวนการ Genetic operation นี้จะเป็นการทำงานแบบขนานกันระหว่างการตัดสลับ (Crossover) และการกลายพันธุ์ (Mutation) ซึ่งจุดนี้โปรแกรมที่เกี่ยวกับ Genetic Algorithm จะมีความแตกต่างกันออกไป ถ้าหากมีการศึกษาในจุดนี้ก็จะสามารถพิสูจน์หารูปแบบของลำดับการทำงานของ Genetic Operation ได้

5.2.2 ในโปรแกรมชุดนี้ได้อนุญาตให้ในแต่ละโครโมโซมสามารถมีความเหมือนกันได้ (ซ้ำกันได้) ดังนั้นจึงมีบางครั้งที่การกระจายตัวของข้อมูลจะน้อยเนื่องจากมี Chromosome ซ้ำกันมาก

นั่นเอง หากศึกษาว่าการซ้ำกันของโครโมโซมนั้นควรมีได้ไม่เกินเท่าไรก็เป็นไปได้ว่าโปรแกรมจะเข้าสู่คำตอบที่ดีได้เร็ว

5.2.3 เพิ่มความสามารถในการเพิ่มจำนวนตัวแปรเนื่องจากปัจจุบันสามารถตั้งค่า population size ได้มากที่สุดเพียง 100 ตัวเท่านั้น แต่ด้วยอัลกอลิทึมที่ผู้เขียนใช้อยู่นี้ไม่สามารถเพิ่มได้มากกว่า 100 ตัว หากมีการพัฒนาอัลกอลิทึมใหม่เช่นแบ่งโปรแกรมเป็น Procedure ย่อยอีก อาจจะทำให้สามารถกำหนด population size ได้มากขึ้น

5.2.4 ในส่วนของการสร้างลำดับและคำนวณนั้นโปรแกรมจะทำงานรวมทั้งประเทศ ทำให้เกิดปัญหาการกระโดดข้ามภาคอย่างมาก ดังนั้นหากสามารถแบ่งโปรแกรมให้คำนวณทีละภาคแล้วค่อยนำมาต่อกันจะสามารถลดปัญหาการกระโดดข้ามภาคลงได้ และยังจะช่วยให้คำตอบนั้นดูสมเหตุผลมากขึ้นด้วย

5.2.5 ในกรณีที่มีการตั้งค่า Crossover and mutation rate รวมกันมากกว่า 100 % ทำให้โปรแกรมจะต้องสุ่มเลือกประชากรออกมาให้เหลือเพียง 100 % ทำให้อาจสูญเสียคำตอบที่ดีๆในการคัดเลือก ดังนั้นจึงควรมีการนำ Roulette Wheel หรือวิธีการอื่นๆมาช่วยในการคัดเลือกในส่วนนี้

5.3 อุปสรรคในการพัฒนา

5.3.1 พบกับเหตุการณ์ Out of Memory ซึ่งในระบบช่วยเหลือของ Visual Basic รายงานดังนี้

Out of memory (Error 7)

More memory was required than is available, or a 64K segment boundary was encountered. This error has the following causes and solutions:

- You have too many applications, documents, or source files open.

  Close any unnecessary applications, documents, or source files that are open.

- You have a module or procedure that's too large.

  Break large modules or procedures into smaller ones. This doesn't save memory, but it can prevent hitting 64K segment boundaries.

- You are running Microsoft Windows in standard mode.

  Restart Microsoft Windows in enhanced mode.

- You are running Microsoft Windows in enhanced mode, but have run out of virtual memory.

  Increase virtual memory by freeing some disk space, or at least ensure that some space is available.

- You have terminate-and-stay-resident programs running.

  Eliminate terminate-and-stay-resident programs.

- You have many device drivers loaded.

  Eliminate unnecessary device drivers.

- You have run out of space for **Public** variables.

  Reduce the number of **Public** variables.

ดังนั้นจึงควรนำอัลกอลิทึมไปปรับปรุงใหม่ให้มีความกระชับกว่าเดิม หรืออาจศึกษาความสามารถของภาษาอื่นๆเพื่อเปรียบเทียบจุดเด่นจุดด้อยและเลือกโปรแกรมภาษาที่เหมาะสม

5.3.2 ไม่สามารถหา method ในการเปลี่ยนฐานข้อมูลที่ทำการเชื่อมต่อไว้แล้วจากคู่มือที่มีขายตามท้องตลาดเนื่องเป็นการทำงานในระดับสูง ทำให้ไม่สามารถเปลี่ยนฐานข้อมูลได้ด้วยตัวโปรแกรมที่พัฒนาขึ้น แต่สามารถเปลี่ยนได้ด้วยการคัดลอกแล้ววาง

บรรณานุกรม

Murata, T., and Ishibuchi, H. "Performance evaluation of genetic algorithms for flow shop scheduling." Problems, Proceedings of the First IEEE Conference on Evolutionary Computation. 2, (1994) : 812-817.

Oliver, I. M., Smith, C. J. and Holland, J. R. C. "A study of permutation crossovers on the traveling salesman problem." Proceedings of the Second International Conference on Genetic Algorithms and their Applications. (1987) : 225- 230.

P.Pongcharoen [et.al.] / int.j. "Production Economics." 78 , (2002) : 311-322.

P.Pongcharoen. "Applying designed experiments to optimize the performance of genetic algorithms used for scheduling complex products in the capital goods industry." Journal of Applied Statistics. 28, (2001) Nos. 3&4 : 441- 455.

Syswerda, G. "Scheduling optimization using genetic algorithm." Handbook of Genetic Algorithm. (1991) : 332- 349.

Tralle, D. "Analyzing of genetic operations in genetic algorithms applied to optimization of manufacturing systems." Undergraduate project, Department of Mechanics, Materials and Manufacturing , Engineering, University of Newcastle. (2000).

ภาคผนวก ก.

การติดตั้งโปรแกรม

เมื่อต้องการติดตั้งโปรแกรมจะต้องเตรียมความพร้อมตามความต้องการของโปรแกรมดังนี้

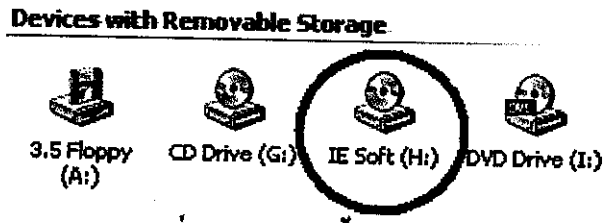
ความต้องการของระบบ (System Requirement)

ตารางที่ ก.1 ความต้องการขั้นต่ำของระบบ

รายการ	ข้อกำหนดขั้นต่ำ	คำแนะนำ
CPU	Pentium II ขึ้นไป	CPU ยิ่งเร็วเท่าไรยิ่งดี
Hard disk	50 MB	60 MB
Ram	64 MB	196 MB
CD-Rom	12 x	ขาดไม่ได้
Monitor	15 "	17" แสดงผลที่ 1,024 x 748
ระบบปฏิบัติการ	Microsoft Window 98,ME,XP	Window XP จะเร็วที่สุด
ระบบจัดการฐานข้อมูล	Microsoft Access 2000	Microsoft Office XP

เริ่มติดตั้งโปรแกรม

1 ใส่แผ่น CD-Rom ของโปรแกรมลงใน Drive CD-Rom และดับเบิ้ลคลิกไอคอน My Computer จะพบ CD-Rom ชื่อ IE Soft ดังรูป



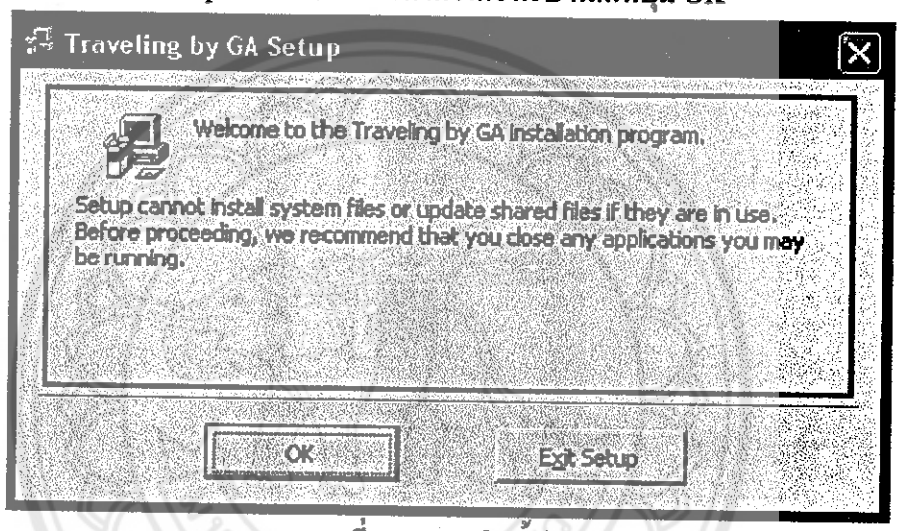
ภาพที่ ก.1 การติดตั้งโปรแกรม 1

(Letter Drive CD-Rom จะแตกต่างกันไปตามจำนวน Drive ของแต่ละเครื่อง)

2 ดับเบิ้ลคลิกที่ CD-Rom ชื่อ IE Soft จะพบกับ File ชื่อ Setup.exe ดังรูป

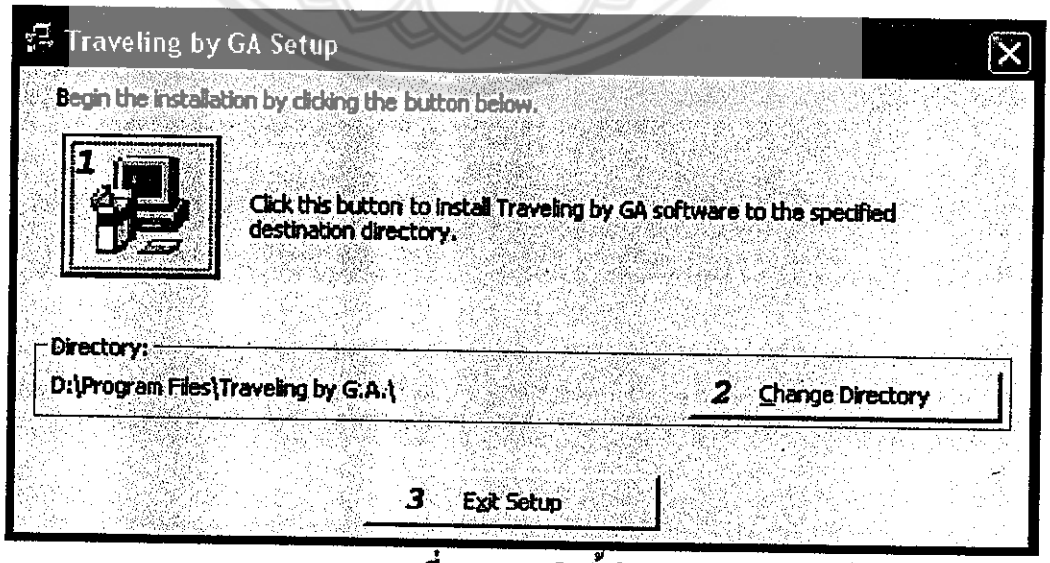
ภาพที่ ก.2 การติดตั้งโปรแกรม 2

3 ดับเบิ้ลคลิกที่ Setup.exe จะพบกับหน้าต่างต้อนรับ คลิกที่ปุ่ม OK



ภาพที่ ก.3 การติดตั้งโปรแกรม 3

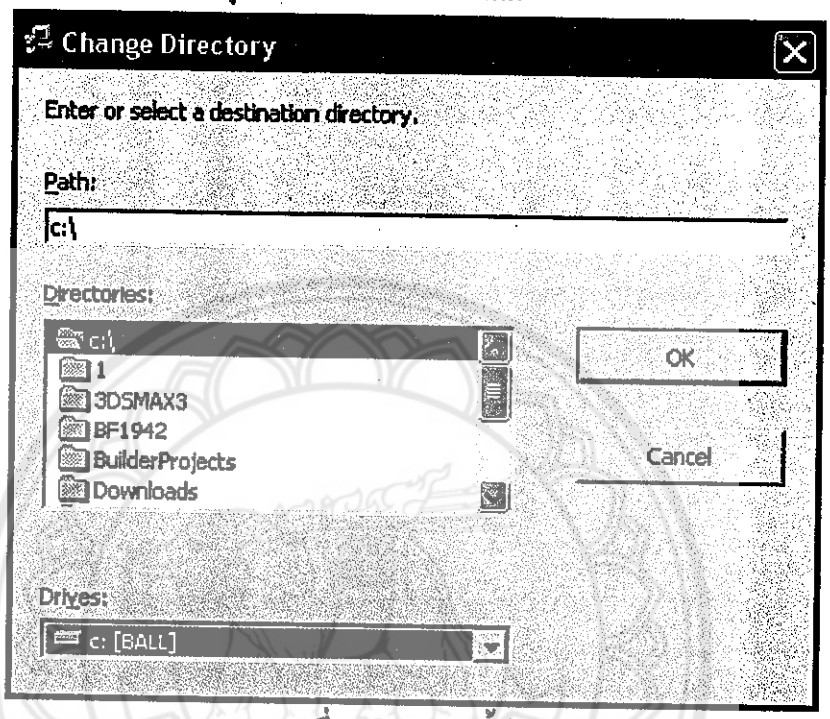
จะพบกับหน้าต่างยืนยันการติดตั้งพร้อมปรับแต่ง



ภาพที่ ก.4 การติดตั้งโปรแกรม 4

หมายเลข 1 คือตกลงติดตั้งโปรแกรมลงใน Directory ที่ปรากฏทางด้านล่าง

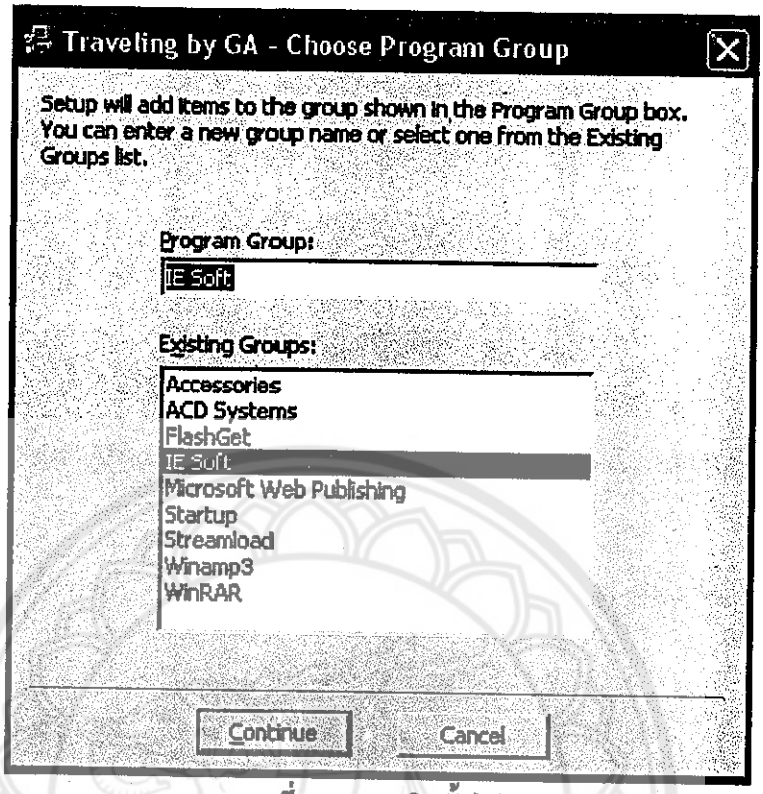
หมายเลข 2 คือการเปลี่ยน Directory ที่จะติดตั้งโปรแกรมจะมี Option ให้เลือกดังรูปด้านล่างนี้เมื่อกำหนดส่วนต่างๆเสร็จแล้วคลิกปุ่ม OK จะกลับไปหน้าเดิม



ภาพที่ ก.5 การติดตั้งโปรแกรม 5

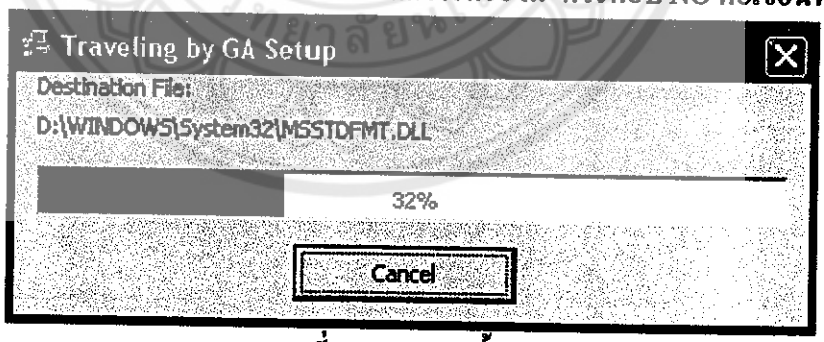
หมายเลข 3 คือปุ่มยกเลิกการติดตั้ง

4 เมื่อคลิกปุ่มตกลงติดตั้ง(หมายเลข 1)จากข้อ3 แล้วจะมีหน้าต่างกำหนดชื่อกลุ่มโปรแกรมเพื่อจะแสดงที่ Start>Program> เมื่อกำหนดชื่อแล้วคลิกปุ่ม Continue



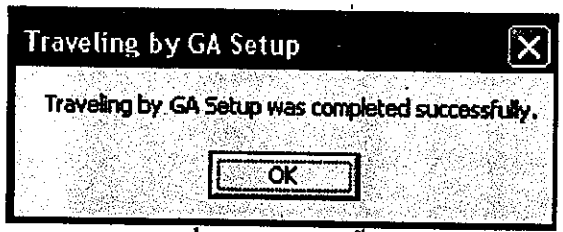
ภาพที่ ก.6 การติดตั้งโปรแกรม 6

5 เริ่มติดตั้งโปรแกรม ในบ้างครั้งอาจจะมีการเขียนทับไฟล์บางไฟล์ซึ่งโปรแกรมจะหยุดถาม ควรอ่านให้ละเอียดและตัดสินใจว่าจะรักษาไฟล์เก่าไว้หรือไม่ ควรตอบ NO คือเขียนทับไปเลย



ภาพที่ ก.7 การติดตั้งโปรแกรม 7

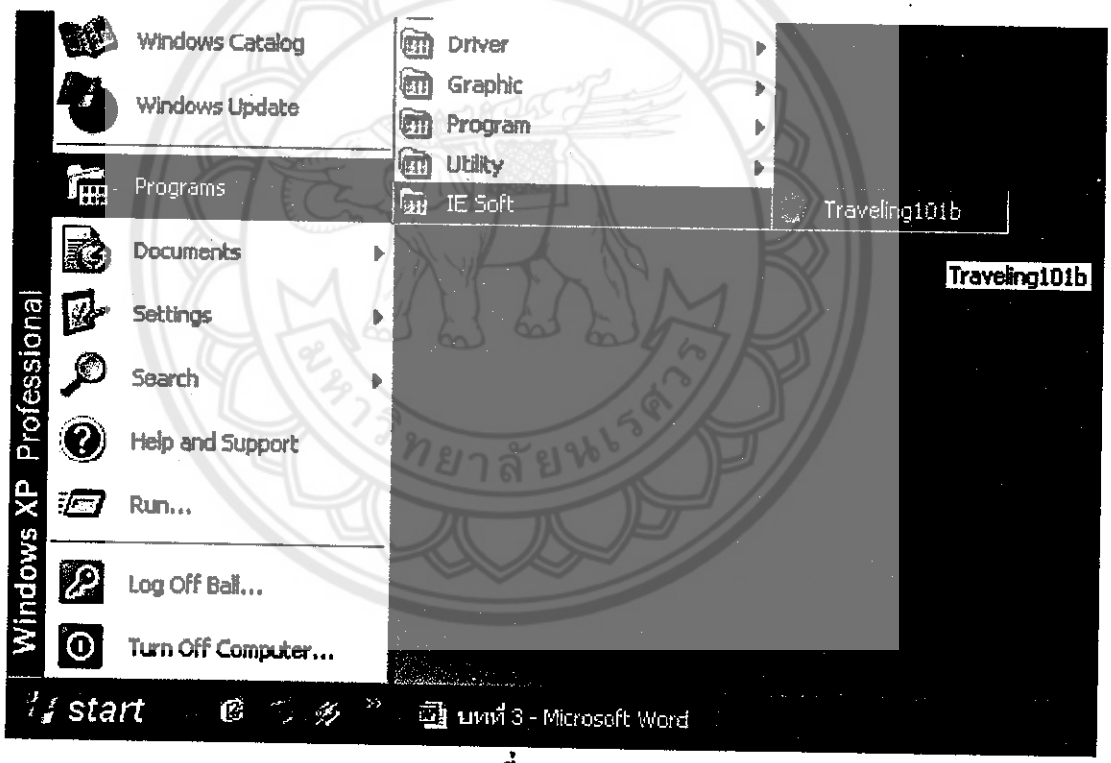
6 ติดตั้งสมบรูณ์คลิกปุ่ม OK เพื่อสิ้นสุดขั้นตอนการติดตั้ง



ภาพที่ ก.8 การติดตั้งโปรแกรม 8

การเรียกใช้งานและการถอนการติดตั้งโปรแกรม

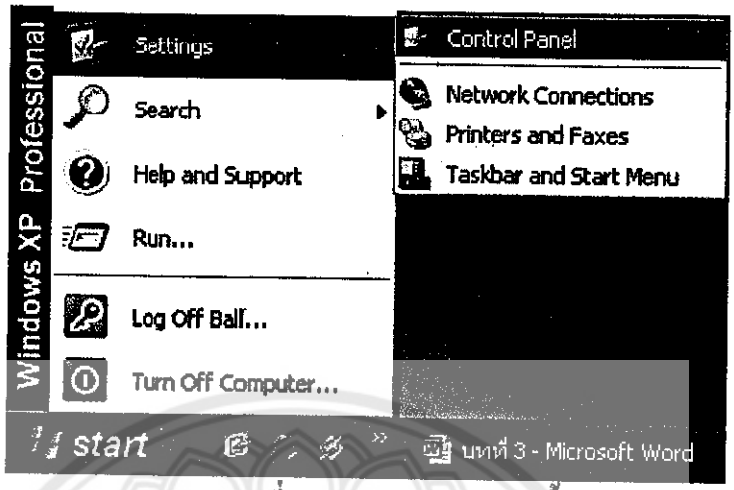
1 การเรียกใช้งาน



ภาพที่ ก.9 การเรียกใช้งาน

สามารถเรียกใช้ได้โดยเลือกที่ Start > Program > IE Soft > Traveling101b

2 การถอนการติดตั้ง



ภาพที่ ก.10 การถอนการติดตั้ง 1

1 เรียก Start > Settings > Control Panel

ภาพที่ ก.11 การถอนการติดตั้ง 2

2 เลือกไอคอน Add or Remove Programs

ภาพที่ ก.12 การถอนการติดตั้ง 3

3 เลือกที่ชื่อ traveling by GA คับเบิลคลิกเลือก Remove

วิธีการใช้งานของโปรแกรม

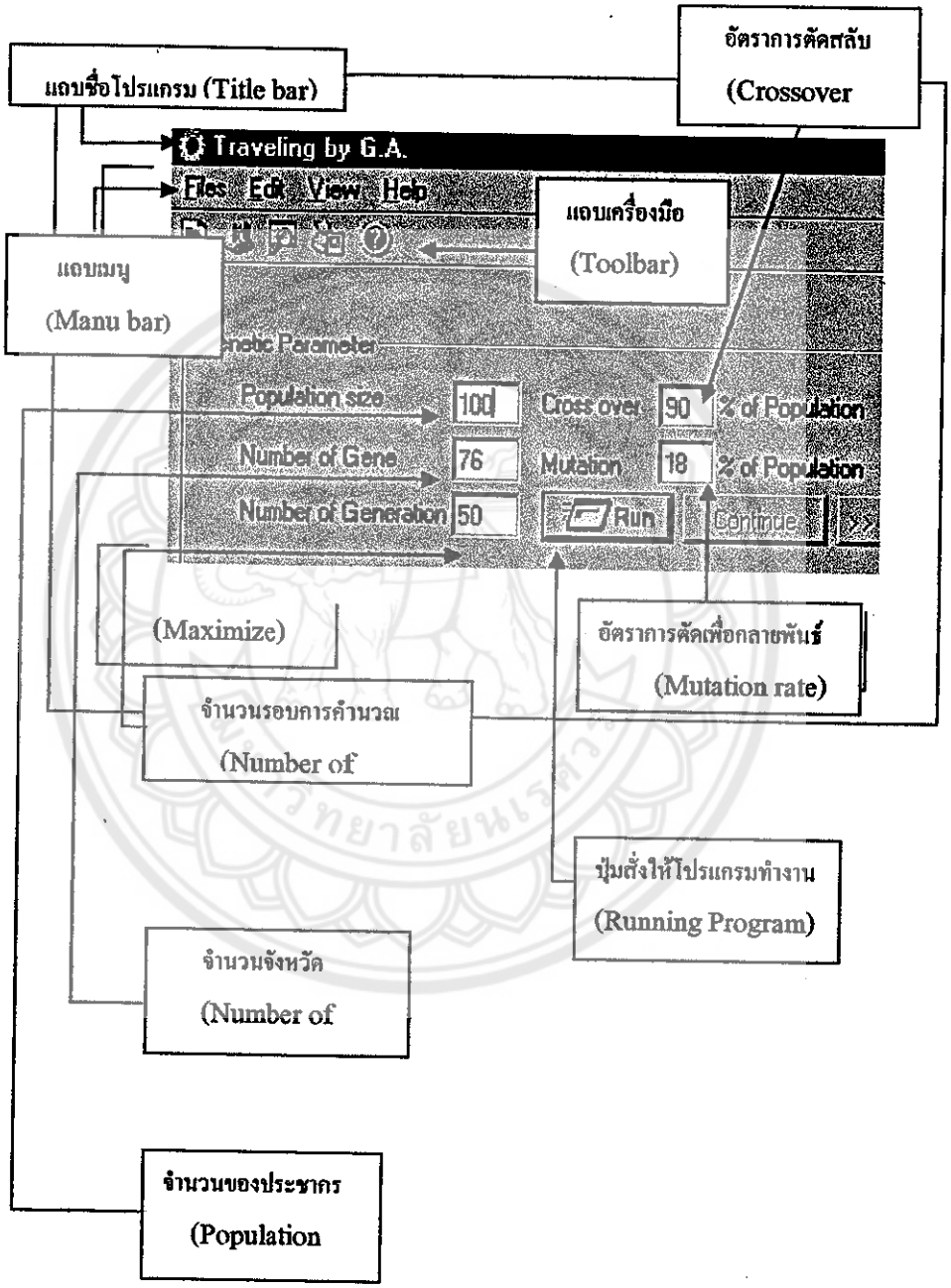
หลักการทำงานถูกแบ่งออกเป็น 2 หน้าจอหลัก

1. ส่วนหน้าจอหลักที่จะทำการตั้งค่าในการรันโปรแกรม

2. ส่วนหน้าจอหลังการคำนวณ จะเป็นการแสดงผลของค่าที่ได้จากการรันโปรแกรม

ส่วนหน้าจอหลัก 1

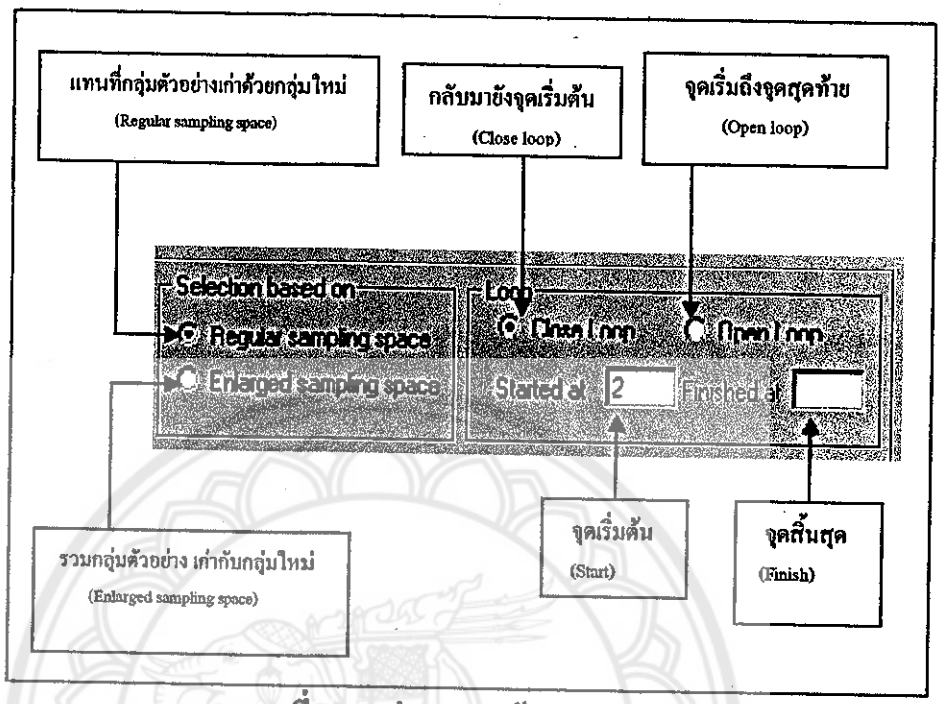
ภาพที่ ก.13 วิธีการใช้งานของโปรแกรม



ส่วนของหน้าจอหลัก 2

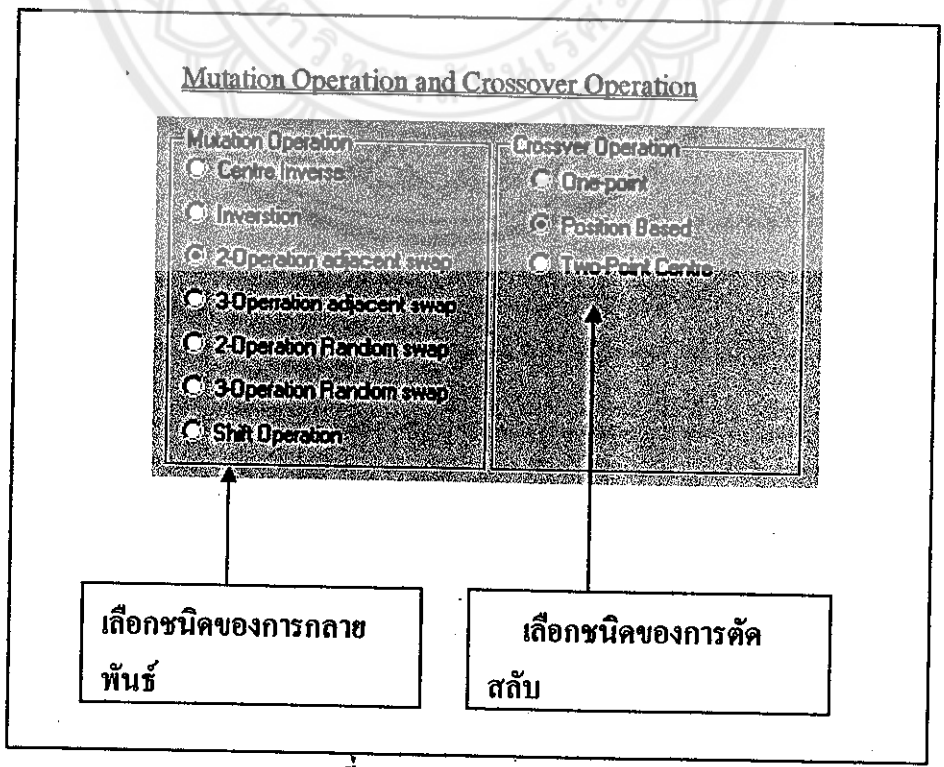
ภาพที่ ก.14 ส่วนของหน้าจอหลัก 2

ส่วนของหน้าจอหลัก 3



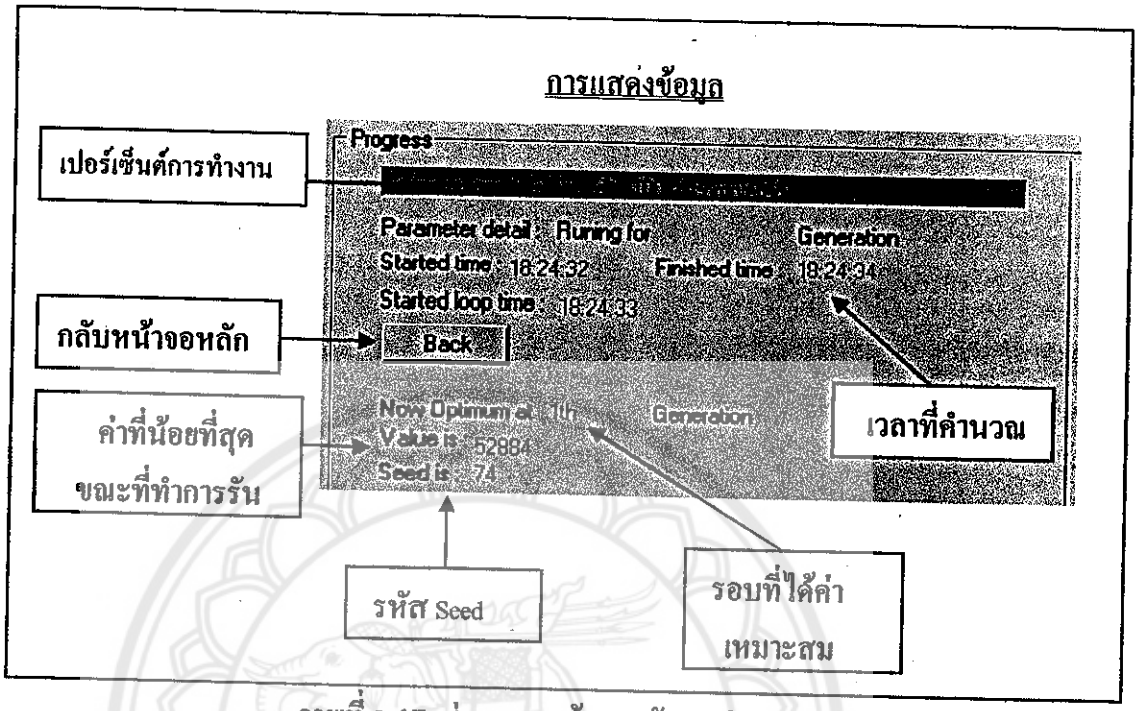
ภาพที่ ก.15 ส่วนของหน้าจอหลัก 3

ส่วนของหน้าจอหลัก 4



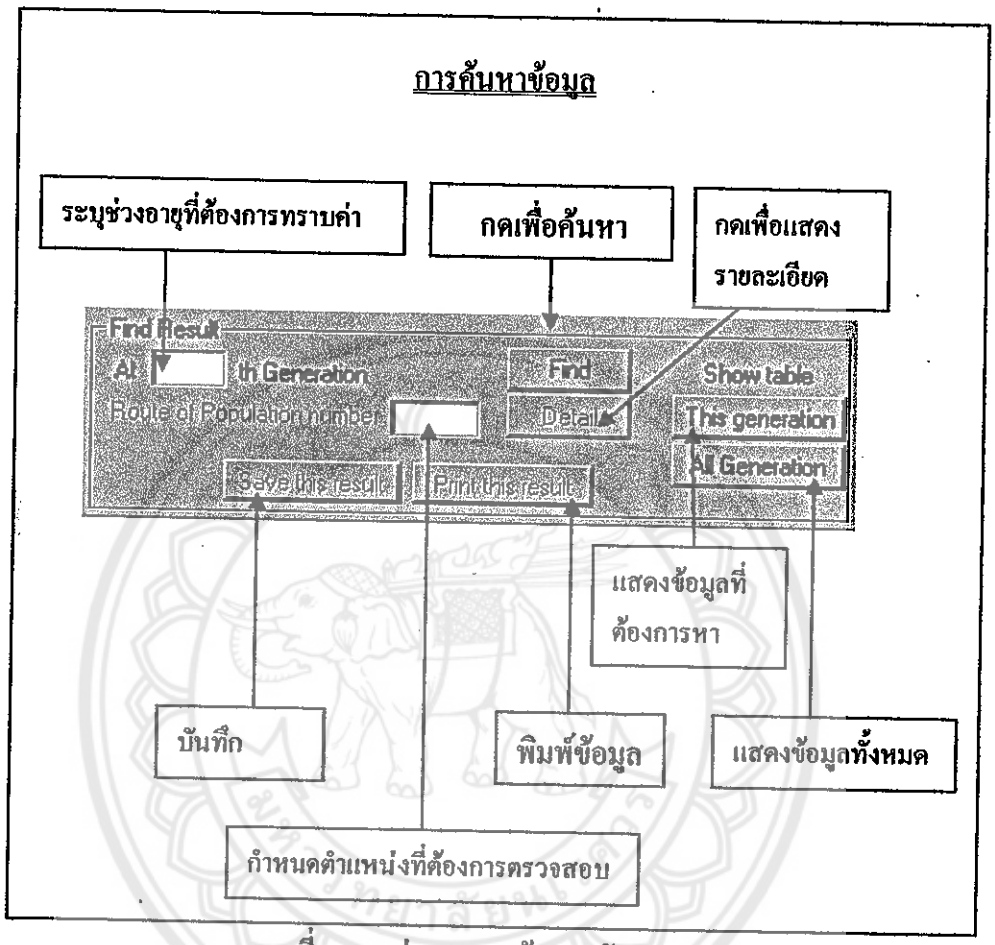
ภาพที่ ก.16 ส่วนของหน้าจอหลัก 4

ส่วนของหน้าจอหลังการคำนวณ 1

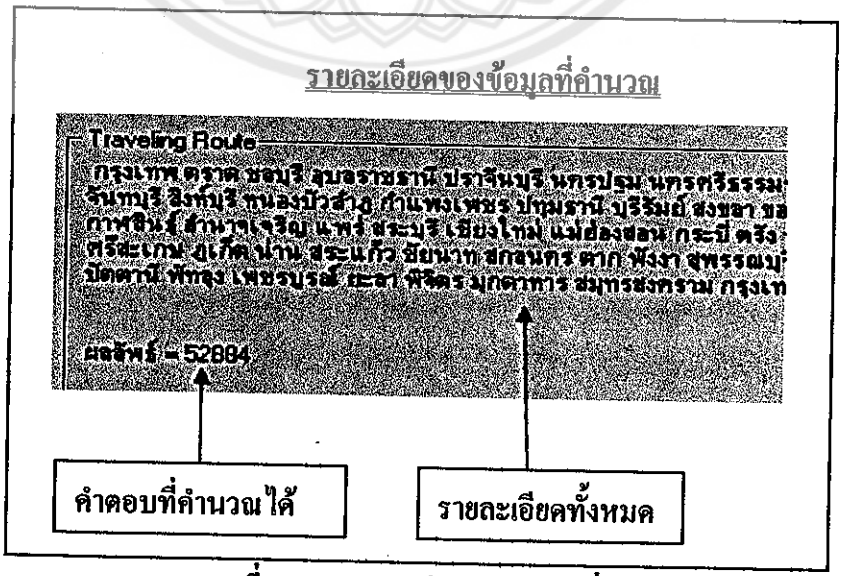


ภาพที่ ก.17 ส่วนของหน้าจอหลังการคำนวณ 1

ส่วนของหน้าจอหลังการคำนวณ 2

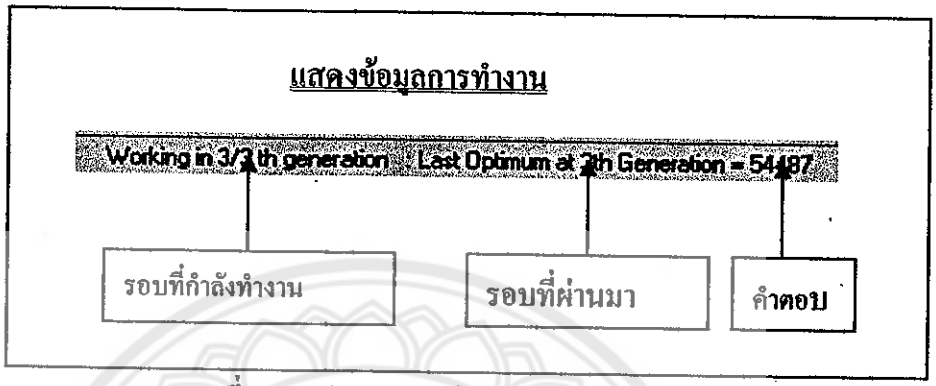


ภาพที่ ก.18 ส่วนของหน้าจอหลังการคำนวณ 2



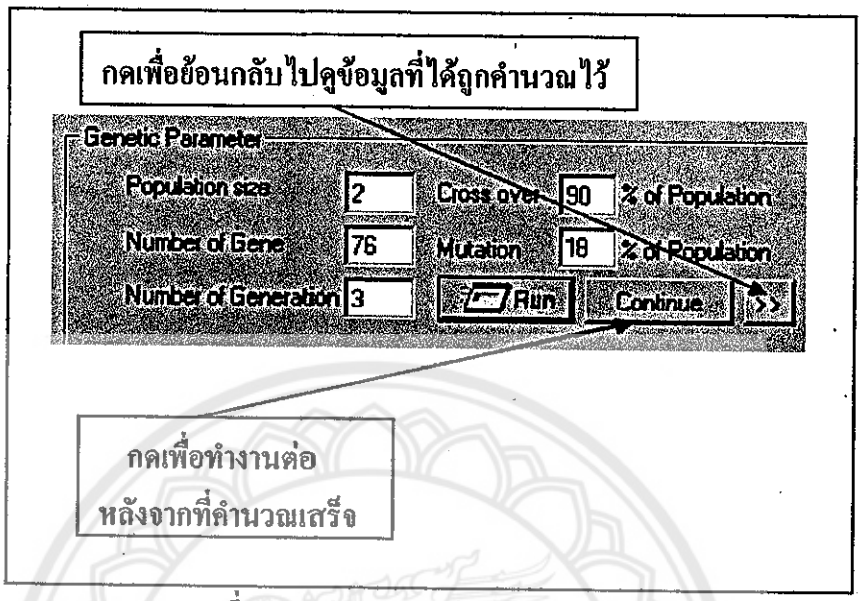
ภาพที่ ก.19 รายละเอียดของข้อมูลที่คำนวณ

ส่วนของหน้าจอหลังการคำนวณ 3



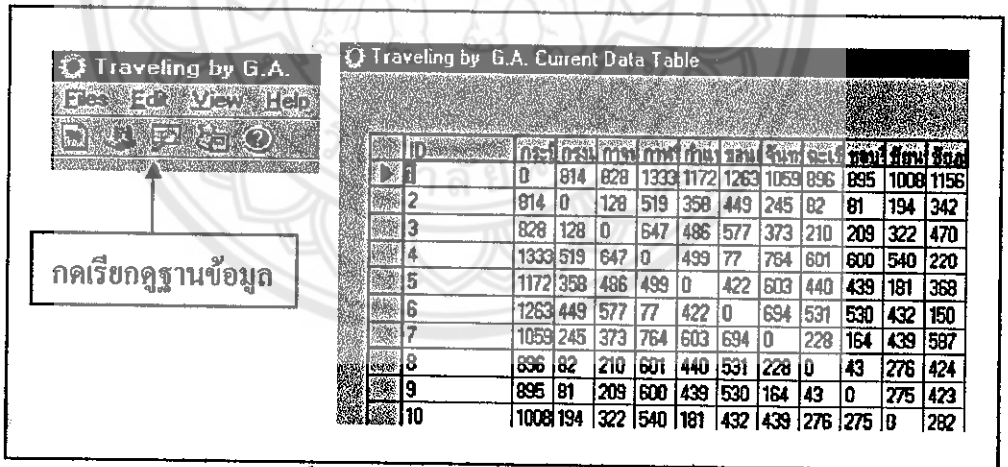
ภาพที่ ก.20 ส่วนของหน้าจอหลังการคำนวณ 3

ส่วนของหน้าจอหลังการคำนวณ 4



ภาพที่ ก.21 ส่วนของหน้าจอหลังการคำนวณ 4

ส่วนของหน้าจอหลังการคำนวณ 5



ภาพที่ ก.22 ส่วนของหน้าจอหลังการคำนวณ 5

ส่วนของหน้าจอหลังการคำนวณ 6

ภาพที่ ก.23 ส่วนของหน้าจอหลังการคำนวณ 6

ส่วนของหน้าจอหลังการคำนวณ 7

การบันทึกข้อมูล

วิธีที่ 1 ใช้คำสั่งที่ Toolbar

- ใช้คำสั่งที่ File เลือก Save หรือ Save as
- ตั้งชื่อ File นามสกุล TXT
- กด OK เพื่อยืนยัน

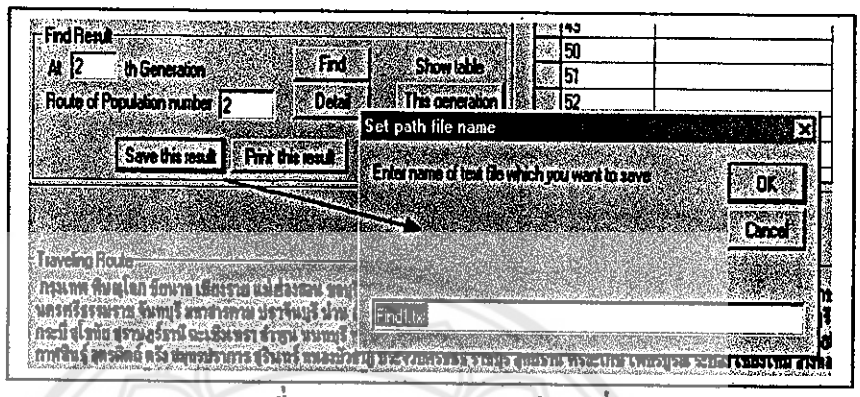
วิธีที่ 2 คีย์ลัด

- กด Ctrl + S หรือ Ctrl + A
- ตั้งชื่อ File นามสกุล TXT
- กด OK เพื่อยืนยัน

วิธีที่ 3 ค้นหาข้อมูลเพื่อทำการบันทึก

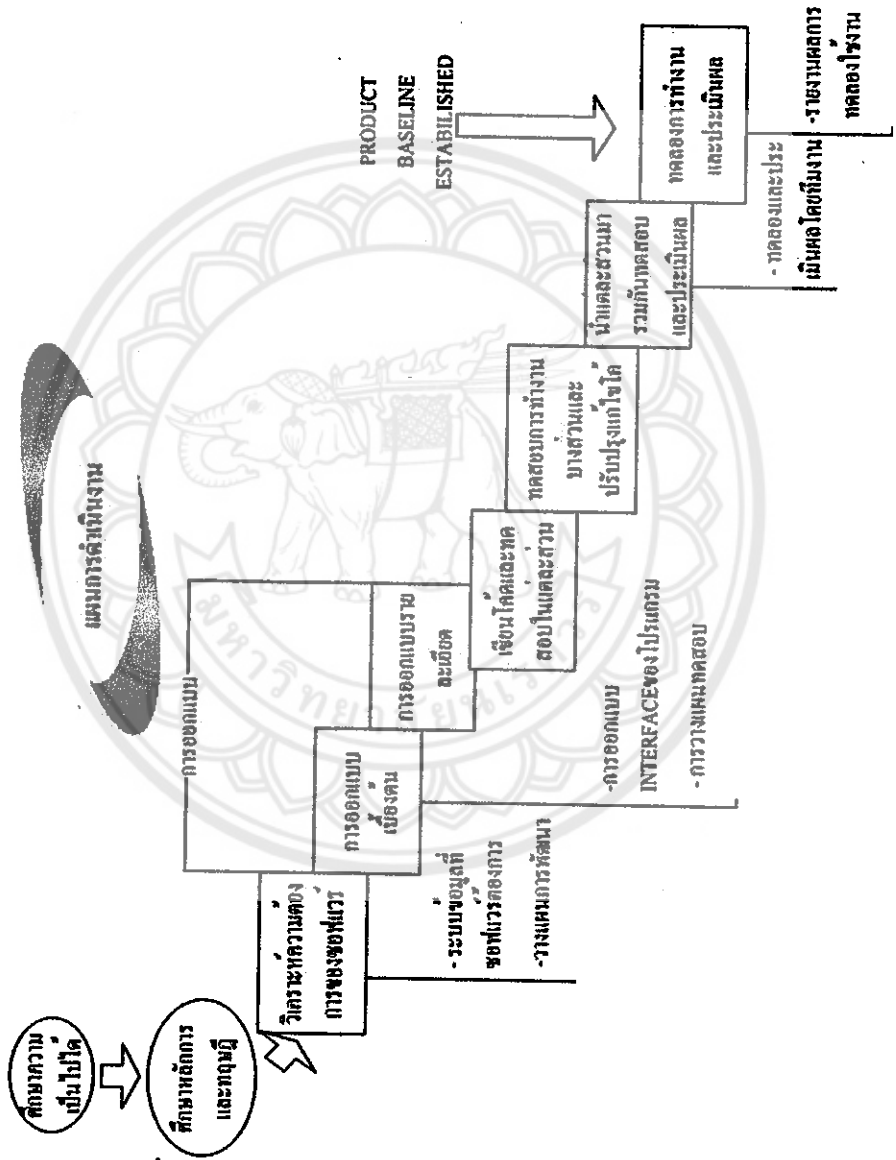
- เมื่อค้นหาข้อมูลเรียบร้อยแล้ว กด Save this result

- ตั้งชื่อ File นามสกุล TXT
- กด OK เพื่อยืนยัน



ภาพที่ ก.24 แสดงการบันทึกวิธีที่ 3

ภาคผนวก ข.



รูปที่ ข.1 แผนภาพแสดงวิธีและขั้นตอนการดำเนินงานวิจัย