

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบบริหารพอร์ตลงทุน

PORTFOLIO MANAGEMENT



นาย นครศ จรรยาไพศาล
นางสาว นีร สุขนวนิช

เลขหมู่.....
เลขทะเบียน..... 61344
วัน,เดือน,ปี..... 17 ก.ค. 2549

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบบริหารพอร์ตลงทุน

PORTFOLIO MANAGEMENT



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2547

ภาควิชาวิศวกรรมคอมพิวเตอร์

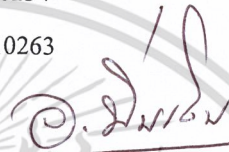
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบบริหารพอร์ตลงทุน

PORTFOLIO MANAGEMENT

ผู้จัดทำ 1. นาย นครธร จรรยาไพศาล รหัสประจำตัว 44010234

2. นางสาว นีร สุชนวนิช รหัสประจำตัว 44010263



อาจารย์ที่ปรึกษา

(รศ.ดร. เอื้อน ปิ่นเงิน)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบบริหารพอร์ตลงทุน

นาย นครศ จรรยาไพศาล 44010234

นางสาว นีอร สุชนวนิช 44010263

รศ.ดร.เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษา

ปีการศึกษา 2547

บทคัดย่อ

รายงานฉบับนี้จะนำเสนอเกี่ยวกับ เว็บ แอปพลิเคชัน ระบบบริหารพอร์ตลงทุน ซึ่งมีความสามารถในการสร้างส่วนผสมของหุ้นกู้ในพอร์ตที่มีความเสี่ยงในการลงทุนน้อยที่สุดและเป็นไปตามความต้องการของนักลงทุน เช่น มีราคาของพอร์ตอยู่ในเกณฑ์ที่ต้องการ มีระยะเวลาการถือครองพอร์ตเท่ากับที่นักลงทุนต้องการ (ในทางการเงินแล้วมักใช้ระยะเวลาการถือครองพอร์ตเป็นเกณฑ์ โดยส่วนผสมของหุ้นกู้ในพอร์ตแต่ละตัวจะถูกสร้างขึ้นจากเซตของหุ้นกู้ทั้งหมดที่นักลงทุนทำการอัปโหลดมา) ซึ่งวิธีการหลักที่แอปพลิเคชันนี้ใช้ในการทำงานคือ เจเนติกอัลกอริทึม โดยวิธีนี้เป็นวิธีการหนึ่งที่จำลองรูปแบบวิธีการทางชีววิทยา ในการให้กำเนิดประชากรรุ่นใหม่หรือขยายเผ่าพันธุ์ในรุ่นลูก uto ต่อๆ ไป โดยอาศัยพื้นฐานความคิดของวิวัฒนาการทางธรรมชาติในการถ่ายทอดลักษณะทางพันธุกรรม เพื่อใช้ในการหาคำตอบที่ดีที่สุดหรือใกล้เคียงที่สุดของปัญหาให้ได้ โดยทั่วไปการใช้เจเนติกอัลกอริทึมนั้น ไม่ได้รับรองว่าจะได้ผลลัพธ์ที่ดีที่สุด แต่เจเนติกอัลกอริทึมจะให้ผลลัพธ์ที่สามารถนำไปใช้งานได้อย่างรวดเร็วและมีประสิทธิภาพ ดังนั้นแอปพลิเคชัน ระบบบริหารพอร์ตลงทุนจึงมีความเหมาะสมในการใช้เจเนติกอัลกอริทึมเพื่อให้ในการหาค่าจำนวนหน่วยที่เหมาะสมให้กับหุ้นกู้แต่ละตัวในส่วนผสมของหุ้นกู้ เนื่องจากเป็นปัญหาที่มีความซับซ้อนและมีคำตอบได้หลากหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกา I คำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Portfolio Management

Nakarate Chanyaphaizarn 44010234

Niorn Suchonwanich 44010263

Assoc.Prof.Dr. Ouen Pinngern Advisor

Academic Year 2547

ABSTRACT

This paper presents a Portfolio Management tool which is the web-based Application that optimize risk for investor's Portfolio. This application is trying to find the most acceptable combination of bonds that meet investor's requirement such as portfolio value requirement and portfolio duration (equal duration is financially a good criteria for Immunization). Combination of bonds is made from uploaded investor's bond set. The main approach which used for this application is Evolutionary Computing called "Genetic Algorithms". Genetic Algorithms are not guaranteed to find the global optimum solution to a problem, but they are generally good at finding "acceptably good" solutions to problems in "acceptably quickly" so this application uses this approach to find appropriate units for each possible combination of bonds and at last use an objective function to find out the most acceptable bond combination for investors. Genetic Algorithms is appropriate for this application because it has an ability to find out a solution even the problem is complicated and dynamic.

สารบัญ

บทคัดย่อภาษาไทย	หน้า I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	VIII

ส่วนที่ 1 บทนำ

บทที่ 1 บทนำ	1
1.1 ปัญหาของการบริหารพอร์ตการลงทุน (Portfolio Management Problem)	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 การดำเนินการพัฒนาโปรแกรมโดยย่อ	2
1.5 ข้อยกเว้นและขอบเขตของการวิจัย	2
1.6 ผลที่คาดว่าจะได้รับ	3

ส่วนที่ 2 ทฤษฎีที่เกี่ยวข้อง

บทที่ 2 ความรู้ทั่วไปเกี่ยวกับตราสารหนี้	4
2.1 ตราสารหนี้ (Bond)	4
2.2 ศัพท์น่ารู้เกี่ยวกับตราสารหนี้ (Glossary)	5
บทที่ 3 เจเนติกอัลกอริทึม	7
3.1 เจเนติกอัลกอริทึมเบื้องต้น	7
3.2 หลักพื้นฐานของเจเนติกอัลกอริทึม	8
3.2.1 ฟังก์ชันเป้าหมายและฟังก์ชันความเหมาะสม	9
3.2.2 รูปแบบโครโมโซม	9
3.2.3 ขั้นตอนการทำงานของ เจเนติกอัลกอริทึม	10
3.2.4 เงื่อนไขที่ใช้ในการหยุดการทำซ้ำ ใน เจเนติกอัลกอริทึม	12
3.2.5 ตัวอย่างการทำงานของเจเนติก อัลกอริทึมและ Roulette Wheel Selection	13

บทที่ 4	การออกแบบ เจเนติกอัลกอริทึมเพื่อแก้ปัญหา	18
4.1	รูปแบบการแทน โครโมโซม	18
4.2	การครอสโอเวอร์ (Crossover)	19
4.2.1	การครอสโอเวอร์แบบดั้งเดิม (Classical Crossover)	19
4.3	การมิวเตชัน (Mutation)	20
บทที่ 5	การปรับปรุงประสิทธิภาพของ เจเนติกอัลกอริทึม	22
5.1	รูปแบบของการคัดเลือกค้นแบบ	22
5.1.1	การคัดเลือกแบบอ้างอิงค่าความเหมาะสม (fitness-based)	23
5.1.1.1	แบบหน้าต่าง (Windowing)	23
5.1.1.2	แบบเชิงเส้น (Linear)	25
5.1.1.3	แบบสัดส่วนเบี่ยงเบนมาตรฐาน (Sigma Truncation)	28
5.1.2	การคัดเลือกแบบอ้างอิงลำดับ (Ranking-based)	29
5.1.2.1	แบบเชิงเส้น (Linear)	30
5.1.2.2	แบบไม่เชิงเส้น (Nonlinear)	31
5.2	วิธีการสุ่มค้นแบบ	32
5.2.1	จำลองแบบการหมุนวงล้อ (Roulette Wheel : RW)	32
5.2.2	สุ่มทศนิยมแบบคืนกลับ	32
5.3	การรีโพรดักชัน (Reproduction)	33
5.3.1	รีโพรดักชันแบบทั่วไป	33
5.3.2	รีโพรดักชันโดยการรักษาโครโมโซมที่ดีที่สุดเอาไว้ (Elitism)	34
5.3.3	รีโพรดักชันโดยรักษาสถานะคงที่แบบชั่ว	34
5.3.4	รีโพรดักชันโดยรักษาสถานะคงที่แบบไม่ชั่ว	35
ส่วนที่ 3	การออกแบบแอปพลิเคชัน	
บทที่ 6	การออกแบบ และพัฒนาแอปพลิเคชัน	36
6.1	อินพุตและเอาต์พุตของแอปพลิเคชัน	36
6.2	โครงสร้างของเนมสเปซ Portfolio Management.	39
6.2.1	เนมสเปซ PortfolioManagement.Bizlogic	39
6.2.2	เนมสเปซ PortfolioManagement.Calculation.	41
6.2.3	เนมสเปซ PortfolioManagement.Data.	41
6.2.4	เนมสเปซ PortfolioManagement.Presentation	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้าน IVรค่า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

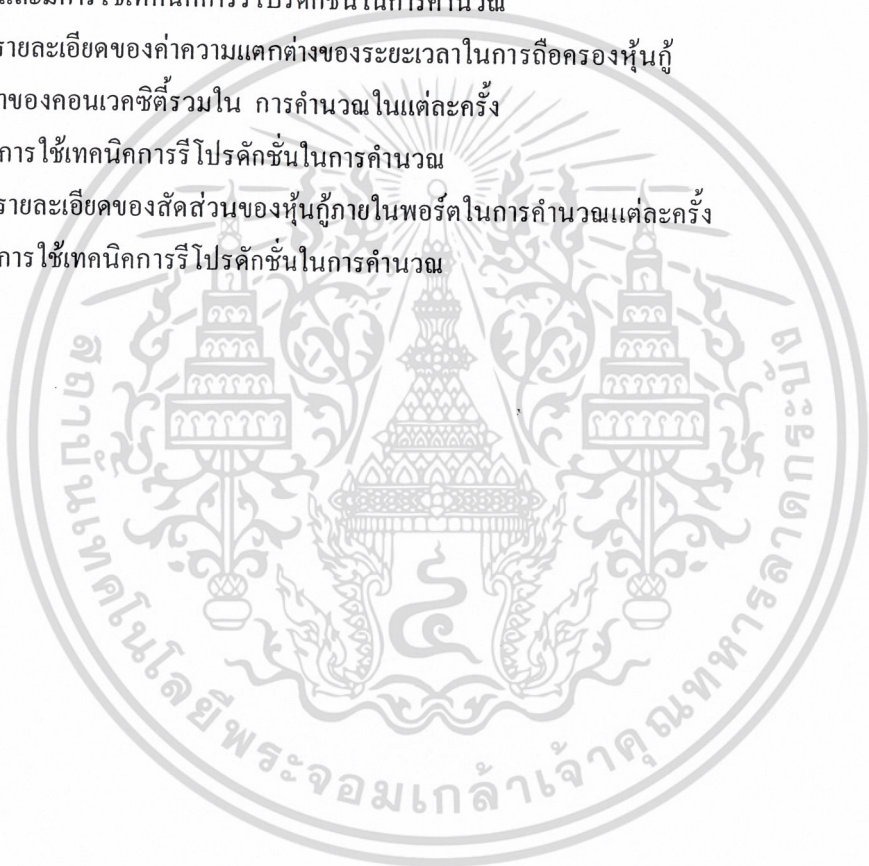
6.3 โครงสร้างของเนมสเปซ GeneticAlgorithm	42
6.3.1 เนมสเปซ GeneticAlgorithm.GeneticStruct	42
6.3.1.1 โครงสร้างของคลาส Chromosome	42
6.3.2 เนมสเปซ GeneticAlgorithm.GeneticBiz	43
6.3.3 เนมสเปซ GeneticAlgorithm.GeneticEquation	45
6.3.4 เนมสเปซ GeneticAlgorithm.GeneticOperator	46
6.3.5 เนมสเปซ GeneticAlgorithm.RoulettWheel.	47
ส่วนที่ 4 บทวิจารณ์ และสรุปผลการทำงานของแอปพลิเคชัน	
บทที่ 7 ผลการทำงานแอปพลิเคชัน	48
ภาคผนวก	57



สารบัญรูป

รูปที่	หน้า
3-1 แสดงหลักการเบื้องต้นของ เจเนติกอัลกอริทึม	7
3-2 แสดงลำดับการทำงานของเจเนติกอัลกอริทึม	12
3-3 แสดง Roulette Wheel ตัวอย่าง	14
4-1 ตัวอย่างโครโมโซม	18
4-2 แสดงการครอสโอเวอร์	19
4-3 แสดงโครโมโซมก่อนทำการครอสโอเวอร์	20
4-4 แสดงการมิวเตชันแบบดั้งเดิม	20
4-5 แสดงการมิวเตชันรูปแบบสุ่มเลือก	21
5-1 การปรับสัดส่วนค่าความเหมาะสมแบบเชิงเส้น	26
5-2 การอ้างอิงลำดับแบบเชิงเส้น	30
5-3 การรีโพรดักชันแบบทั่วไป	33
5-4 การรีโพรดักชันโดยการรักษาโครโมโซมที่ดีที่สุดเอาไว้	34
5-5 การรีโพรดักชันโดยการรักษาสถานะแบบซ้ำ	34
5-6 การรีโพรดักชันโดยการรักษาสถานะแบบไม่ซ้ำ	35
6-1 แสดงหน้าอินพุทของแอฟฟลิเคชัน	36
6-2 แสดงหน้าอินพุทของแอฟฟลิเคชันหลังจากมีการอัปเดตข้อมูล	37
6-3 แสดงหน้าเอาต์พุทหน้าที่ 1	37
6-4 แสดงหน้าเอาต์พุทหน้าที่ 2	38
6-5 แสดงหน้าเอาต์พุทหน้าที่ 3	38
6-6 แสดงหน้าเอาต์พุทหน้าที่ 4	39
6-7 แสดงโครงสร้างของคลาส Bizlogic	40
6-8 แสดงโครงสร้างของคลาส BondData	41
6-9 แสดงรายละเอียดของคลาส Chromosome	42
6-10 แสดงโครงสร้างของคลาส GeneticBiz	44
6-11 แสดงโครงสร้างของคลาส GeneticEquation	45
6-12 แสดงโครงสร้างของคลาส GeneticOperator	46
6-13 แสดงขั้นตอนการทำงานของแอฟฟลิเคชันระบบบริหารพอร์ตลงทุน	47

7-1 แสดงรายละเอียดของการใส่ค่าอินพุทของแอปพลิเคชัน	48
7-2 แสดงเอาต์พุตที่เป็นรายละเอียดของหุ้นกู้ที่นักลงทุนสนใจ	49
7-3 แสดงรายละเอียดของการคำนวณในแต่ละครั้ง	51
7-4 แสดงรายละเอียดของค่าความแตกต่างของระยะเวลาในการถือครองหุ้นกู้ และค่าของคอนเวกซ์ตีฟรอมใน การคำนวณในแต่ละครั้ง	52
7-5 แสดงรายละเอียดของสัดส่วนของหุ้นกู้ภายในพอร์ตในการคำนวณแต่ละครั้ง	52
7-6 แสดงรายละเอียดของการคำนวณ ในแต่ละครั้งโดยระยะเวลาในการถือครองพอร์ตมีค่าเท่ากับ 1.1 ปี และมีการใช้เทคนิคการรีโปรคักชันในการคำนวณ	54
7-7 แสดงรายละเอียดของค่าความแตกต่างของระยะเวลาในการถือครองหุ้นกู้ และค่าของคอนเวกซ์ตีฟรอมใน การคำนวณในแต่ละครั้ง และมีการใช้เทคนิคการรีโปรคักชันในการคำนวณ	55
7-8 แสดงรายละเอียดของสัดส่วนของหุ้นกู้ภายในพอร์ตในการคำนวณแต่ละครั้ง และมีการใช้เทคนิคการรีโปรคักชันในการคำนวณ	55



สารบัญตาราง

ตาราง	หน้า
3-1 แสดงรูปแบบการแทนไครโมโซมด้วย binary string ขนาด 4 บิต	13
3-2 แสดงประชากรไครโมโซม และค่าความเหมาะสมของไครโมโซมในเจเนอเรชันแรก	13
3-3 แสดงประชากรไครโมโซมในเจเนอเรชันแรก	15
3-4 แสดงค่าความเหมาะสมของไครโมโซมในเจเนอเรชันแรก	15
3-5 แสดงค่าความน่าจะเป็นของแต่ละไครโมโซมที่จะถูกเลือก	16
5-1 แสดงการเปรียบเทียบค่าความเหมาะสมจากการปรับสัดส่วนและไม่ปรับสัดส่วน กับปัญหาแบบหาค่าใช้จ่ายมากที่สุด	24
5-2 แสดงการเปรียบเทียบค่าความเหมาะสมจากการปรับสัดส่วนและไม่ปรับสัดส่วน กับปัญหาแบบหาค่าใช้จ่ายน้อยที่สุด	24
5-3 แสดงค่าความเหมาะสม โดยใช้วิธีการปรับค่าแบบเชิงเส้น	27
5-4 แสดงค่าความเหมาะสมเฉลี่ย โดยใช้วิธีการปรับค่าแบบเชิงเส้น โดยมี ไครโมโซมตัวหนึ่งมีค่าความเหมาะสมน้อยเกินไป	28
5-5 แสดงค่าต่างๆ โดยใช้วิธีปรับสัดส่วนแบบตัดส่วนเบี่ยงเบนมาตรฐาน	29
5-6 แสดงการเปรียบเทียบการอ้างอิงค่าความเหมาะสมกับการอ้างอิงลำดับค่าความเหมาะสม	31
5-7 แสดงการสุ่มเลือกไครโมโซมต้นแบบโดยวิธีสุ่มทศนิยมแบบคืนกลับ	32
5-8 แสดงการสุ่มเลือกไครโมโซมต้นแบบโดยวิธีหมุนวงล้อด้วยค่าทศนิยมของค่าความคาดหวัง	33

บทที่ 1

บทนำ

1.1 ปัญหาของการบริหารพอร์ตการลงทุน (Portfolio Management Problem)

ในปัจจุบันนี้การลงทุนโดยการซื้อขายหุ้นและตราสารหนี้นั้นได้รับความนิยมเพิ่มมากขึ้นอย่างต่อเนื่อง ซึ่งหุ้นก็ถือเป็นหุ้นอีกรูปแบบหนึ่งที่ได้รับคามนิยมเพิ่มขึ้น โดยในประเทศไทยนั้นตลาดหุ้นกู้หรือ ตลาดตราสารหนี้ภาคเอกชนในประเทศไทยต้องถือว่ายังอยู่ในช่วงเริ่มต้น แต่ก็มีอัตราการเติบโตอย่างต่อเนื่อง โดยการลงทุนในหุ้นกู้นั้นมีประโยชน์ ได้แก่

- อัตราผลตอบแทน หุ้นกู้เสนออัตราผลตอบแทนที่สูงกว่าการลงทุนในพันธบัตรรัฐบาลที่มีลักษณะและอายุคงเหลือใกล้เคียงกัน
- เป็นแหล่งรายได้ประจำ เนื่องจากหุ้นกู้และตราสารหนี้จ่ายดอกเบี้ยเป็นงวดๆ และจ่ายต้นเงินคืนด้วย
- ความมั่นคงปลอดภัยของการลงทุน
- สามารถซื้อขายเปลี่ยนมือได้

โดยก่อนที่จะทำการลงทุนนั้นนักลงทุนต้องมีความเข้าใจกับตัวแปร เงื่อนไข และปัจจัยต่างๆ ที่มีผลกระทบต่อมูลค่าและราคาตราสารหนี้ อัตราดอกเบี้ย และอัตราผลตอบแทน เพื่อจะเลือกหุ้นกู้ในการลงทุนให้ได้ประโยชน์อย่างสูงสุด ซึ่งปัจจัยต่างๆ เหล่านี้มีอยู่มากมายจึงเป็นการยากสำหรับนักลงทุนสมัครเล่นที่ต้องการลงทุนกับหุ้นกู้ที่จะทำความเข้าใจได้ทั้งหมด จึงเป็นที่มาของ เว็บแอปพลิเคชัน “Portfolio Management” ที่จะช่วยนักลงทุนในการวิเคราะห์และเลือกหุ้นกู้เพื่อที่จะลงทุนได้อย่างมีประสิทธิภาพ และได้พอร์ตการลงทุนที่เป็นไปตามความต้องการของนักลงทุนด้วย

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาเทคนิคทางด้านปัญญาประดิษฐ์ และแนวทางในการนำเทคนิคมาประยุกต์ใช้ กับปัญหาทางธุรกิจ
2. การสร้างเว็บแอปพลิเคชันที่มีความสามารถในการช่วยนักลงทุนตัดสินใจเลือกหุ้นกู้เข้าพอร์ตลงทุน เพื่อให้พอร์ตนั้นมีคุณสมบัติเป็นไปตามความต้องการของนักลงทุน

1.3 ขอบเขตของโครงการ

คือ การพัฒนาแอปพลิเคชัน “Portfolio Management” ซึ่งเป็น Web-Based Applicationจนเสร็จสมบูรณ์ โดยใช้โปรแกรมต่างๆ ในการพัฒนาแต่ละส่วนของแอปพลิเคชันดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1) Microsoft Visual Studio .NET ซึ่งจะใช้ภาษา C# ในการพัฒนา โดยจะใช้ในการเขียนโปรแกรมส่วนที่เป็น

- หน้ารับ Input จากผู้ใช้
- หน้าแสดงผลที่ได้จากการคำนวณของโปรแกรม
- ส่วนที่ทำหน้าที่คำนวณค่าต่างๆซึ่งเป็นคุณสมบัติของหุ้นกู้
- ส่วนที่ทำหน้าที่ค้นหาพอร์ตที่ดีที่สุดโดยใช้เจเนติกอัลกอริทึมในการค้นหา

(2) Easy Chart_275 และ Java Script

- ใช้เพื่อแสดงผลในรูปแบบกราฟ

แอปพลิเคชันนี้ จะใช้ได้กับตลาดหุ้นกู้ในประเทศไทยเท่านั้น เนื่องจากในการคำนวณบางส่วนต้องใช้ข้อมูลจาก ฐานข้อมูลของสถาบันการเงินในประเทศไทย ซึ่งจะมีเพียงข้อมูลของหุ้นกู้ที่จดทะเบียนในประเทศไทยเท่านั้นด้วย

1.4 การดำเนินการพัฒนาโปรแกรมโดยย่อ

เว็บแอปพลิเคชัน “Portfolio Management” นี้เป็นการนำความรู้และเทคนิคของปัญญาประดิษฐ์ (Artificial Intelligence) มาใช้เพื่อช่วยนักลงทุนในการเลือกหุ้นเข้าพอร์ตเพื่อให้ได้พอร์ตที่มีคุณสมบัติตามความต้องการของผู้ลงทุน เช่น ได้พอร์ตที่ให้กระแสเงินสด (Cashflow) ตามที่ต้องการในแต่ละช่วงปี โดยจะต้องเป็นพอร์ตที่สร้างความเสี่ยงให้กับนักลงทุนน้อยที่สุดและในขณะเดียวกันก็ สร้างผลตอบแทนให้กับนักลงทุนมากที่สุด (เทียบกับเงินที่นักลงทุนใช้ในการลงทุน) ซึ่งการพัฒนาโปรแกรมนี้ประกอบด้วยเนื้อหา 4 ส่วนคือ

บทที่ 2 นำเสนอความรู้เบื้องต้นเกี่ยวกับตราสารหนี้และศัพท์ที่จำเป็นต้องรู้เพื่อจะสามารถใช้งานแอปพลิเคชัน

บทที่ 3 นำเสนอการศึกษาทฤษฎีและการทำงานของ เจเนติกอัลกอริทึม

บทที่ 4 นำเสนอการออกแบบเจเนติกอัลกอริทึมเพื่อแก้ปัญหา การบริหารพอร์ตลงทุน โดยจะนำเสนอตั้งแต่รูปแบบการแทน โครโมโซม การครอสโอเวอร์และการมิวเตชัน

บทที่ 5 นำเสนอเทคนิคที่ใช้ในการปรับปรุงประสิทธิภาพของเจเนติกอัลกอริทึมได้ดีขึ้น เช่น การ รีโพรดักชัน รูปแบบต่างๆ

บทที่ 6 นำเสนอขั้นตอนการพัฒนา ออกแบบ แอปพลิเคชัน

บทที่ 7 นำเสนอผลการทำงานของแอปพลิเคชัน

1.5 ข้อจำกัดและขอบเขตของการวิจัย

ขอบเขตของการวิจัยนี้ คือ จะทำการสร้างระบบบริหารพอร์ตลงทุนที่จะนำมาใช้งานในตลาดหุ้นกู้ประเทศไทยเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ผลที่คาดว่าจะได้รับ

1. แอปพลิเคชัน “Portfolio Management” จะเป็นแอปพลิเคชันที่เป็นประโยชน์อย่างมากสำหรับนักลงทุนที่มีความต้องการเข้ามาลงทุนในตลาดหุ้นของไทย
2. ได้รับความรู้ในการพัฒนาแอปพลิเคชันที่แก้ไขปัญหาทางสังคมศาสตร์และธุรกิจ ซึ่งเป็นปัญหาที่มีความแปรผันสูง (Dynamic) โดยใช้เทคนิคและความสามารถทางด้านปัญญาประดิษฐ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ความรู้ทั่วไปเกี่ยวกับตราสารหนี้

2.1 ตราสารหนี้ (Bond)

ตราสารหนี้ [1] คือ ตราสารทางการเงินที่แสดงถึงความเป็นหนี้ระหว่างกัน โดยลักษณะของการเป็นตราสารเป็นสิ่งที่สามารถเปลี่ยนมือได้ กล่าวอีกนัยหนึ่งตราสารหนี้ก็คือการกู้ยืมเงินชนิดหนึ่งที่มีความเป็นมาตรฐาน ผู้ออกตราสารเป็นผู้กู้หรือลูกหนี้ ในขณะที่ผู้ให้กู้เป็นผู้ซื้อหรือเจ้าหนี้ โดยทั้งสองฝ่ายมีข้อผูกพันกันทางกฎหมายที่จะได้รับชำระเงิน หรือผลประโยชน์อื่นใด เช่น ดอกเบี้ย เงินต้น ตามเวลาที่กำหนด ทั้งนี้ตราสารหนี้มีคุณสมบัติที่สามารถแบ่งเป็นหน่วยย่อยๆ ที่เท่ากันได้ โดยผลประโยชน์ หรืออัตราผลตอบแทนเท่ากันทุกหน่วย และมีคุณสมบัติที่สามารถซื้อขายเปลี่ยนมือกันได้จนกว่าจะหมดอายุของตราสารนั้นๆ

เนื่องจากคำนิยามของตราสารหนี้มีได้มีอยู่ในเอกสารทางกฎหมาย หรือเอกสารของหน่วยราชการที่เป็นผู้ควบคุมดูแลการออกตราสารหนี้ คำอธิบายข้างต้นจึงเป็นการประมวลจากลักษณะ และวิธีการตลอดจนกฎหมายที่เกี่ยวข้อง โดยหากพิจารณานิยามของตราสารหนี้ตามพระราชบัญญัติหลักทรัพย์ และตลาดหลักทรัพย์ พ.ศ. 2535 มีเพียงการให้คำนิยามของคำว่าหุ้นกู้ ไว้ดังนี้

“หุ้นกู้ หมายความว่า ตราสารแห่งหนึ่งไม่ว่าจะเรียกชื่อใดที่แบ่งเป็นหน่วย แต่ละหน่วยมีมูลค่าเท่ากัน และกำหนดประโยชน์ตอบแทนไว้เป็นการล่วงหน้าในอัตราเท่ากันทุกหน่วย โดยบริษัทออกให้แก่ผู้ให้กู้ยืมเงินหรือผู้ซื้อ เพื่อแสดงสิทธิที่จะได้รับเงินหรือผลประโยชน์อื่นของผู้ถือตราสารดังกล่าว แต่ไม่รวมถึงตัวเงิน”

สำหรับการเรียกชื่อตราสารหนี้ นั้น มักมีความแตกต่างกันเล็กน้อยระหว่างในประเทศไทยและต่างประเทศ โดยทั่วไปในต่างประเทศใช้คำว่า Bond สำหรับตราสารหนี้ทั้งที่ออกโดยรัฐบาล และบริษัทเอกชนซึ่งอาจมีหลักทรัพย์ค้ำประกัน (Secured bond) หรือไม่มีหลักทรัพย์ค้ำประกัน (Unsecured bond) ก็ได้ โดยกรณีที่ใช้คำว่า Debenture มักใช้ในกรณีที่ไม่มีการค้ำประกัน สำหรับในประเทศไทยนิยมเรียกตราสารหนี้ที่ออกโดยรัฐบาล หรือองค์กรของรัฐว่า พันธบัตร ส่วนตราสารหนี้ที่ออกโดยภาคเอกชนจะเรียกว่า หุ้นกู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ศัพท์น่ารู้เกี่ยวกับตราสารหนี้ (Glossary)

Basis Point (bp)	หน่วยย่อยของอัตราผลตอบแทนโดย 1 bp มีค่าเท่ากับ 0.01%
Bond	พันธบัตร หรือหุ้นกู้ คือ ตราสารแห่งหนึ่งระยะยาวซึ่งผู้ออกมีข้อผูกพันตามกฎหมายที่จะชำระดอกเบี้ย และเงินต้นแก่ผู้ซื้อตามเวลาที่กำหนดในต่างประเทศจะใช้คำว่า Bond
Convexity	เป็นคุณสมบัติอย่างหนึ่งของตราสารหนี้ซึ่งชี้ให้เห็นว่าความสัมพันธ์ระหว่างราคากับอัตราผลตอบแทนไม่ได้เป็นเส้นตรง โดยในกรณีที่เป็น Option-free bond และในกรณีที่ลักษณะของ Positive Convexity ดังนั้น เมื่ออัตราดอกเบี้ยลดลงราคาตราสารหนี้จะเพิ่มขึ้นในสัดส่วนที่มากกว่าการลดลงของราคาที่เกิดขึ้น เมื่ออัตราดอกเบี้ยสูงขึ้น
Corporate Bond	หุ้นกู้ภาคเอกชน หมายถึง ตราสารหนี้ที่ภาคเอกชนเป็นผู้ออกเพื่อเป็นการระดมทุน
Coupon Rate	อัตราดอกเบี้ยที่ระบุไว้ของตราสารหนี้โดยกำหนดเป็นอัตราร้อยละต่อปี
Default	การผิดนัดชำระหนี้เมื่อครบกำหนดชำระ หรือการไม่สามารถทำตามเงื่อนไข หรือข้อตกลงที่กำหนดเอาไว้ในสัญญาเงินกู้ (Covenant)
Duration	เครื่องมืออย่างหนึ่งที่ใช้วัดระดับความเสี่ยงในการลงทุนในตราสารหนี้มี 2 ความหมายคือ 1) Macaulay Duration คือ ระยะเวลาโดยเฉลี่ยซึ่งผู้ถือตราสารหนี้จะได้รับเงิน (เงินต้น และดอกเบี้ย) คืน 2) Modified Duration คือ ตัวเลขที่ชี้ให้เห็นถึงอัตราการเปลี่ยนแปลงของราคาของตราสารหนี้เมื่อมีการเปลี่ยนแปลงของอัตราผลตอบแทน (Yield to Maturity)
Face Value (Par value)	มูลค่าที่ตราเอาไว้ต่อหน่วย คือ มูลค่าที่กำหนดไว้บนหน้าค้ำของตราสารหนี้แต่ละหน่วยส่วนใหญ่ คือหน่วยละ 1,000 บาท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Government Bond	พันธบัตรภาครัฐ หมายถึง ตราสารหนี้ระยะยาวที่ออกโดยรัฐบาล
Junk Bond	ตราสารหนี้ซึ่งได้รับการจัดอันดับความน่าเชื่อถือต่ำกว่าระดับที่ยอมรับเพื่อการลงทุน
Present Value	มูลค่าปัจจุบันของกระแสเงินในอนาคต
Time to Maturity	ระยะเวลาที่เหลืออยู่จนถึงวันครบกำหนดอายุของตราสารหนี้
Treasury Bill	ตั๋วเงินคลัง คือตราสารหนี้ทางการเงินระยะสั้นไม่เกิน 12 เดือนที่รัฐบาลเป็นผู้ออกจำหน่ายโดยวิธีการประมูล และชำระเงินในราคาตราหักส่วนลด เมื่อครบกำหนด ผู้ถือกรรมสิทธิ์จะได้รับเงินเต็มจำนวนตามราคาตรา
Value	มูลค่า หรือปริมาณของการซื้อขายซึ่งแสดงเป็นจำนวนเงิน
Zero-coupon Bond	ตราสารหนี้ที่กำหนดให้ไม่มีการจ่ายดอกเบี้ย(coupon) ในช่วงอายุของตราสาร โดยทั่วไปจะออกจำหน่ายในราคาที่ต่ำกว่ามูลค่าที่ตราเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

เจเนติกอัลกอริทึม

ปัจจุบันนี้มีปัญหาหลากหลายรูปแบบ ที่ต้องการค้นหาคำตอบที่ดีที่สุด (Optimal solution) เช่น ปัญหาทางวิศวกรรม ปัญหาทางด้านธุรกิจ ซึ่งคำตอบที่ดีที่สุดเหล่านั้น สามารถหาคำตอบได้จากหลายวิธี โดยการเลือกวิธีเพื่อใช้หาคำตอบนั้นมักจะแตกต่างกันไปตามชนิดของปัญหา ซึ่งเทคนิคหนึ่งที่มีความนิยมในการแก้ปัญหาเพื่อค้นหาคำตอบที่ดีที่สุดนั้นก็คือ การใช้กฎเกณฑ์ทางธรรมชาติมาช่วยในการวิจัย เช่น นิวรอน เน็ตเวิร์ค (Neural Network) ฟัซซี่ลอจิก (Fuzzy Logic) และ เจเนติกอัลกอริทึม (Genetic Algorithms) เป็นต้น

เจเนติกอัลกอริทึม เป็นวิธีการหนึ่งที่จำลองรูปแบบวิธีการทางชีววิทยา ในการให้กำเนิดประชากรรุ่นใหม่หรือขยายเผ่าพันธุ์ในรุ่นลูก ต่อๆ ไปโดยอาศัยพื้นฐานความคิดของวิวัฒนาการทางธรรมชาติในการถ่ายทอดลักษณะทางพันธุกรรม เพื่อใช้ในการหาคำตอบที่ดีที่สุดหรือใกล้เคียงที่สุดของปัญหาให้ได้

3.1 เจเนติกอัลกอริทึมเบื้องต้น

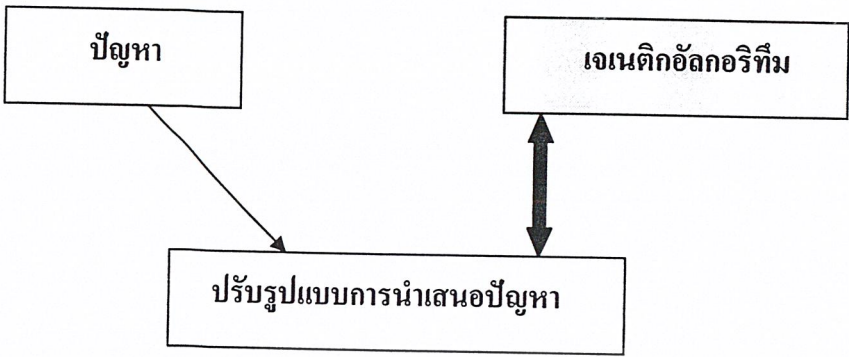
ปี ค.ศ. 1975 John Holland[2] ได้ศึกษาทฤษฎีวิวัฒนาการทางธรรมชาติ (Natural Evolution) ในการกำเนิดประชากร (Population) ในรุ่นต่อๆ ไป โดยกระบวนการธรรมชาติทางชีววิทยานั้นประกอบด้วย

(1) การคัดเลือกทางธรรมชาติ (Natural Selection) ซึ่งก็คือ สิ่งมีชีวิตใดแข็งแรงกว่าย่อมมีโอกาสได้อยู่รอดมากกว่า หมายถึง สิ่งมีชีวิตที่มีโครโมโซมซึ่งประกอบด้วยยีนที่มีลักษณะดี จะมีโอกาสอยู่รอดได้มากกว่า ซึ่งโครโมโซมที่สามารถอยู่รอดได้ดีกว่า ก็จะมีโอกาสที่จะถ่ายทอดยีนที่ดีไปให้ลูกหลานได้มากกว่าเช่นกัน

(2) กระบวนการทางพันธุศาสตร์ (Genetic Operation) คือการให้กำเนิดโครโมโซมใหม่โดยการผสมพันธุ์ โดยการครอสโอเวอร์ หรือการกลายพันธุ์จากมิวเตชัน

Holland ได้ทำการวิจัยโดยจำลองแบบเพื่อทดลองกับปัญหาแบบต่างๆ จุดมุ่งหมายที่จะศึกษาระบบการปรับปรุงตัวเอง (self-adaptive process) และเพื่อสร้างโปรแกรมระบบผู้เชี่ยวชาญ (artificial system software) เพื่อแก้ปัญหา โดยอาศัยแนวความคิดของระบบทางธรรมชาติ ซึ่งเป็นจุดเริ่มต้นให้ค้นพบวิธีการใหม่ซึ่งเรียกว่า เจเนติกอัลกอริทึม (Genetic Algorithms : เจเนติกอัลกอริทึม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3-1 แสดงหลักการเบื้องต้นของ เจเนติกอัลกอริทึม

รูปที่ 3-1 แสดงหลักการเบื้องต้นในการใช้เจเนติกอัลกอริทึมแก้ปัญหา โดยในตอนแรกเมื่อมีการกำหนดปัญหาแล้ว จะต้องมีการปรับปรุงรูปแบบของปัญหาให้มีลักษณะที่เหมาะสมเพื่อที่จะนำเสนอให้กับเจเนติกอัลกอริทึม ค่าย

เจเนติกอัลกอริทึม นั้นเป็นวิธีการค้นหาคำตอบโดยอาศัยวิธีการเลียนแบบการคัดเลือกทางธรรมชาติและกระบวนการทางพันธุศาสตร์ ซึ่งได้แก่การรวมหรือ สลับเปลี่ยนแปลงตัวแปรต่างๆ ที่เป็นโครงสร้างของปัญหาเพื่อให้ได้คำตอบที่ต้องการ โดยการรวมและสลับเปลี่ยนนี้จะใช้หลักการสุ่ม เพื่อใช้ปรับปรุงให้ได้คำตอบที่ดีขึ้น การค้นหาคำตอบจากรุ่นหนึ่งไปยังรุ่นถัดไปตามวิวัฒนาการของธรรมชาติ นั้น คำตอบในรุ่นใหม่จะเกิดจากการสร้างความสัมพันธ์ระหว่างโครงสร้างของตัวแปรที่เหมาะสมในรุ่นก่อน ทำให้ได้คำตอบที่ดีขึ้น จะเห็นได้ว่าถึงแม้ วิธีการพื้นฐานของ เจเนติกอัลกอริทึมจะเป็นวิธีการแบบสุ่ม แต่การสุ่มนั้นก็ยังมีหลักการและประสิทธิภาพ เนื่องจาก

- เจเนติกอัลกอริทึมค้นหาคำตอบภายใต้โครงสร้างของปัญหาที่ได้รับการกำหนดรูปแบบแล้ว (coding)
- เจเนติกอัลกอริทึมค้นหาคำตอบโดยพิจารณาจากประชากรคำตอบ หรือกลุ่มคำตอบ ไม่ใช่การพิจารณาคำตอบใดคำตอบหนึ่ง
- เจเนติกอัลกอริทึมค้นหาคำตอบโดยอ้างอิงจาก ฟังก์ชันเป้าหมายของปัญหา เพื่อค้นหาความเหมาะสมของแต่ละคำตอบจากกลุ่มคำตอบนั้น

3.2 หลักพื้นฐานของ เจเนติกอัลกอริทึม

ความฉลาด [3] คือ ความสามารถของระบบที่สามารถปรับเปลี่ยนพฤติกรรมให้เท่าทันกับสภาพแวดล้อมที่มีการเปลี่ยนแปลงไปได้ อีโวลูชันนารีคอมพิวเตอร์ (Evolutionary Computation) คือ การใช้คอมพิวเตอร์เพื่อที่จะนำเสนอการปรับเปลี่ยนนั้น ซึ่งเรียกการปรับเปลี่ยนนั้นได้อีกอย่างหนึ่งว่า วิวัฒนาการ (Evolution) โดยผลของการจำลองการทำงานนี้จะมีการทำซ้ำไปเรื่อยๆ เพื่อปรับปรุงคุณภาพของคำตอบ (solution) และจะหยุดการทำซ้ำเมื่อได้คำตอบ (solution) ที่ดีที่สุด หรือเป็นคำตอบ (solution) ที่เป็นไปได้มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีโวลูชันนารีแมทชีนเลิร์นนิง (Evolutionary Machine learning) นี้ มีพื้นฐานจาก โมเดล การคำนวณ (computational model) ของการคัดเลือกตามธรรมชาติและพันธุกรรมศาสตร์ โดย คำว่า อีโวลูชันนารีคอมพิวเตอร์ (Evolutionary Computation) นั้นจะครอบคลุมทั้ง เจเนติกอัลกอริทึม และ เจเนติกโปรแกรมมิ่ง (genetic programming) ซึ่งเทคนิคที่ใช้ในการนำเสนอ นั้นได้แก่ การคัดเลือกตามธรรมชาติ มิวเตชัน และ การรีโพรดักชัน (Reproduction)

3.2.1 ฟังก์ชันเป้าหมายและฟังก์ชันความเหมาะสม

เจเนติกอัลกอริทึมมีพื้นฐานอยู่บนผลลัพธ์จากการหาคำตอบที่ผ่านมา วิธีการของเจเนติก อัลกอริทึมจะไม่พิจารณาขั้นตอนของการแก้ปัญหา แต่จะพิจารณาว่าคำตอบใหม่ที่ได้รับดีขึ้นหรือไม่ หรือ เป็นคำตอบที่ใกล้เคียงคำตอบที่ต้องการหรือไม่ โดยสามารถดูความใกล้เคียงนั้นได้จาก ฟังก์ชันเป้าหมาย (Objective Function : f) ซึ่งเป็นฟังก์ชันที่แสดงความสัมพันธ์ของแต่ละตัวแปร สำหรับฟังก์ชัน ความเหมาะสม (Fitness Function : F) เป็นฟังก์ชันที่กำหนดความเหมาะสม(fitness) ของแต่ละ โครโมโซมเปรียบเสมือนเป็นค่าความสามารถในการอยู่รอดของแต่ละ โครโมโซม นั่นคือเป็นฟังก์ชันที่ใช้ กำหนดโอกาสของโครโมโซม หรือบ่งบอกความเหมาะสมที่แต่ละ โครโมโซมจะถูกคัดเลือกขึ้นมา มากน้อยเพียงใด

โดยทั่วไปแล้วมักใช้ฟังก์ชันเป้าหมายเป็นฟังก์ชันความเหมาะสมเลย หรืออาจใช้ฟังก์ชัน เป้าหมายที่ถูกปรับให้เหมาะสมกับการนำเสนอ เจเนติกอัลกอริทึม เป็นฟังก์ชันความเหมาะสมก็ได้

3.2.2 รูปแบบโครโมโซม

วิวัฒนาการทางธรรมชาตินั้น เกิดจากความเปลี่ยนแปลงต่างๆ ของสิ่งมีชีวิตที่เกิดขึ้นใน โครโมโซม ดังนั้นจุดเริ่มต้นของการจำลองรูปแบบทางธรรมชาติของเจเนติกอัลกอริทึมเพื่อใช้แก้ปัญหา จึงเริ่มจากการมองปัญหาเทียบเท่ากับโครโมโซมตัวหนึ่งที่ประกอบด้วยยีนต่างๆซึ่งหมายถึงลำดับข้อมูลที่จะแปลความหมายแล้วสามารถให้คำตอบปัญหาค่าหนึ่ง การมองยีนของเจเนติกอัลกอริทึมนั้นให้ถือ เสมือนยีนทางพันธุกรรมที่ แสดงความหมายหรือเป็นตัวแทนคำตอบตัวหนึ่ง ในทางพันธุศาสตร์นั้น ยีน จะเป็นตัวบ่งบอกลักษณะความอยู่รอดในสภาวะแวดล้อมขณะนั้น นอกจากนี้การมองยีนในเจเนติก อัลกอริทึมนั้นยีนหมายถึง ตัวแปร พารามิเตอร์ เงื่อนไข ที่เป็นองค์ประกอบของปัญหา ดังนั้นการกำหนด รูปแบบของโครโมโซมของแต่ละปัญหาอาจทำได้โดย การแปลงตัวแปร พารามิเตอร์ เงื่อนไข หรือ ข้อกำหนดต่างๆให้อยู่ในรูปลำดับของยีนบนโครโมโซม หรือเรียกว่า สตริง (String)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ขั้นตอนการทำงานของ เจเนติกอัลกอริทึม

เจเนติกอัลกอริทึม เป็นวิธีการค้นหาคำตอบโดยใช้วิธีการทางสถิติ ที่มีพื้นฐานมาจาก การวิวัฒนาการทางชีววิทยา เจเนติกอัลกอริทึมจะสามารถทำงานก็ต่อเมื่อได้รับทราบปัญหาที่ต้องการแก้ และไบนารี สตริง (binary string) ซึ่งมีโอกาสที่จะเป็นคำตอบ ก่อนที่จะเริ่มกระบวนการตามขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 : เปลี่ยนตัวแปรซึ่งอยู่ในปัญหาที่ต้องการแก้ให้อยู่ในรูปของโครโมโซม ซึ่งมีขนาดคงที่ เลือกขนาดของประชากรโครโมโซม (Chromosome population) N ค่าความน่าจะเป็นในการครอสโอเวอร์ p_c ค่าความน่าจะเป็นในการเกิด มิวเตชัน p_m

ขั้นตอนที่ 2 : เลือกฟังก์ชันความเหมาะสมที่จะใช้ในการวัดประสิทธิภาพ หรือความเหมาะสมของแต่ละ โครโมโซม ในโดเมนปัญหานั้น โดยฟังก์ชันความเหมาะสมจะเป็นตัวเลือกโครโมโซม ที่จะนำไปเข้าสู่การรีโพรดักชัน ต่อไป

ขั้นตอนที่ 3 : ทำการสร้างโครโมโซมให้มี population เท่ากับ N โดยใช้วิธีสุ่มเลือก

$$x_1, x_2, \dots, x_n$$

ขั้นตอนที่ 4 : นำโครโมโซมทุกตัวมาเข้าฟังก์ชันความเหมาะสมเพื่อหาค่าความเหมาะสมของโครโมโซมแต่ละตัว

$$f(x_1), f(x_2), \dots, f(x_n)$$

ขั้นตอนที่ 5 : เลือกโครโมโซมหนึ่งคู่จากทั้งหมดที่มีอยู่เพื่อนำมาเป็นโครโมโซมต้นแบบ (parent chromosome) ซึ่งโครโมโซมต้นแบบนี้จะถูกเลือกโดยดูจากค่าความเหมาะสมที่ได้มาจากฟังก์ชันความเหมาะสม ซึ่งโครโมโซมที่มีค่าความเหมาะสมมากกว่า จะมีความน่าจะเป็นที่จะถูกเลือกมาเข้าสู่มากกว่า โครโมโซม ตัวที่มีค่าความเหมาะสมน้อย

ขั้นตอนที่ 6 : สร้างโครโมโซมลูก(offspring) 1 คู่ ในอีกรุ่น (generation) หนึ่ง จากโครโมโซมต้นแบบโดย ใช้ ตัวดำเนินการทางพันธุศาสตร์ (genetic operator) คือ ครอสโอเวอร์ และมิวเตชัน

ขั้นตอนที่ 7 : นำโครโมโซมในรุ่นใหม่ไปไว้ในอีกประชากรโครโมโซม หนึ่ง (new chromosome population)

ขั้นตอนที่ 8 : ทำ ขั้นตอนที่ 5 ซ้ำ จนกระทั่งขนาดของประชากรโครโมโซมรุ่นใหม่ (new chromosome population) เท่ากับ ขนาดของประชากรโครโมโซมเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 9 : เลือกโครโมโซมในรุ่นใหม่ๆ เพื่อใช้เป็นต้นแบบแทนโครโมโซมต้นแบบคู่ก่อน

ขั้นตอนที่ 10 : กลับไปยัง ขั้นตอนที่ 4 และ ทำกระบวนการทั้งหมด ซ้ำไปเรื่อยๆ จนกระทั่งได้คำตอบที่เหมาะสม

การทำงานของเจเนติก อัลกอริทึม สามารถเขียนเป็น **pseudo code** การทำงานได้ดังนี้
BEGIN

```

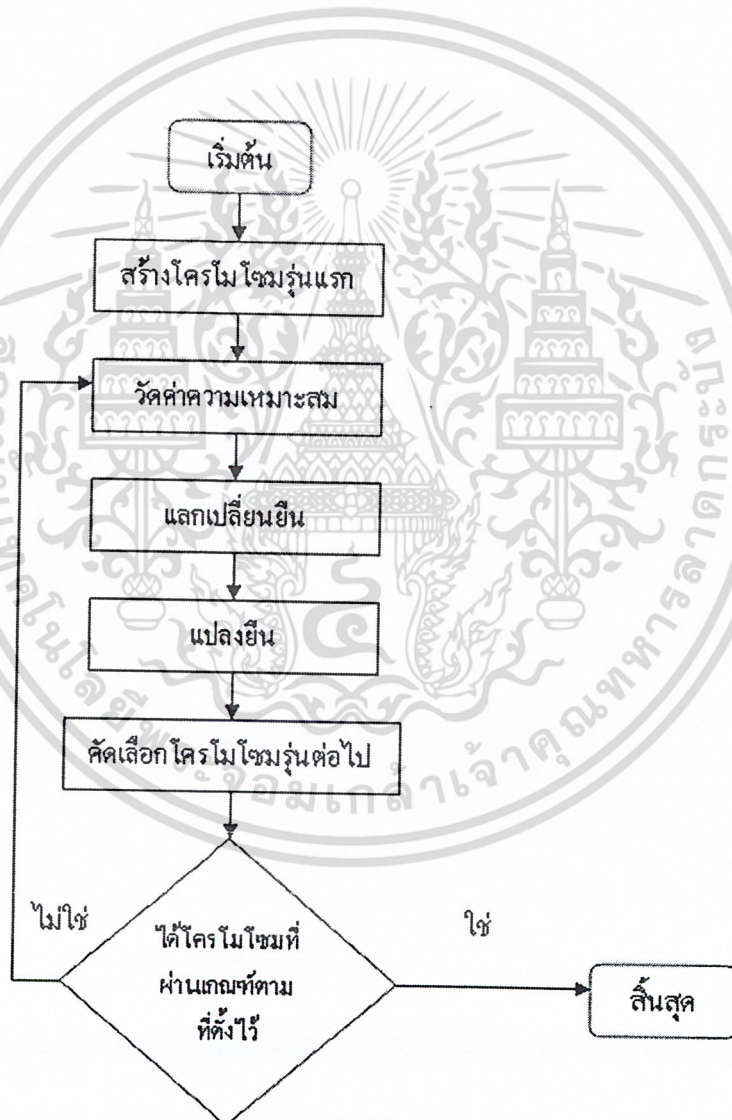
T:=0;
//สร้างประชากรต้นกำเนิดโดยการสุ่ม
Initpopulation P(t);
//วิเคราะห์ความเหมาะสมของแต่ละโครโมโซมประชากรต้นกำเนิด
Evaluate P(t);
//ตรวจสอบเงื่อนไขความพอใจ เช่น ค่าความเหมาะสม
while not terminate
begin
T:=T+1;
//คัดเลือกโครโมโซมต้นแบบจากประชากรรุ่นก่อน
P(t) :=Selectparents P(T-1);
//แลกเปลี่ยนส่วนยีนภายในโครโมโซมต้นแบบ
Recombine P(T);
//มิวเตชันโครโมโซมต้นแบบ
Mutation P(T);
//วิเคราะห์ความเหมาะสมของประชากรรุ่นใหม่
Evaluate P(T);
//ประชากรรุ่นใหม่หลายเป็นประชากรรุ่นเก่าต่อไป
P'(T) := P(T);
End;
END.

```

จากขั้นตอนการทำงานของ เจเนติกอัลกอริทึม จะเห็นได้ว่ากระบวนการทำงานจะเป็นกระบวนการแบบทำซ้ำ (iteration) โดยแต่ละครั้งที่มีการทำซ้ำจะเรียกว่า “เจเนอเรชัน” (generation) ซึ่งจำนวนการทำซ้ำ ของ เจเนติกอัลกอริทึมอย่างง่าย จะมีตั้งแต่ 50 ถึง 500 ครั้ง [4](Mitchell,1996) กลุ่มของเจเนอเรชันทั้งหมดเรียกว่า “รัน” (run) โดยความคาดหวังของเจเนติก อัลกอริทึม อยู่ที่เมื่อจบการ “รัน” แล้ว โครโมโซม ที่เหมาะสมจะเป็นคำตอบอย่างน้อย 1 ตัว

3.2.4 เจียนไซที่ใช้ในการหยุดการทำซ้ำ ใน เจเนติกอัลกอริทึม

เนื่องจากเจเนติกอัลกอริทึมเป็นวิธีการค้นหาคำตอบ โดยใช้หลักการทางสถิติทำให้ในบางครั้งระดับความเหมาะสมของโครโมโซม ในแต่ละเจเนอเรชันนั้นมีความเปลี่ยนแปลงไม่มากนักก่อนที่จะพบโครโมโซมที่มีระดับความเหมาะสมมากพอที่จะเป็นคำตอบที่เหมาะสมได้ ซึ่งปัญหาที่ตามมาก็คือ จะหยุดการทำซ้ำเมื่อใด เจียนไซที่จะใช้หยุดการทำรีโพรดักชัน (reproduction) นั่นคือ จะหยุดการทำซ้ำเมื่อได้ทำรีโพรดักชัน (reproduction) จนครบจำนวนเจเนอเรชันที่ได้กำหนดไว้แล้ว และจะทำการเลือกโครโมโซมที่ดีที่สุดประชากรโครโมโซมมาเพื่อเป็นคำตอบ ซึ่งถ้าไม่มีโครโมโซมที่มีความเหมาะสมมากพอที่จะเป็น คำตอบได้ การทำงานของเจเนติก อัลกอริทึมก็จะเริ่มขึ้นอีกครั้ง ตามรูปที่ 3-2



รูป 3-2 แสดงลำดับการทำงานของเจเนติกอัลกอริทึม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 ตัวอย่างการทำงานของเจเนติก อัลกอริทึมและ Roulette Wheel Selection

กรณีศึกษาที่ 1 หาค่ามากที่สุดของฟังก์ชัน $15x - x^2$ ซึ่งพารามิเตอร์ x มีค่าระหว่าง $0 - 15$ และตั้งสมมติฐานว่า x เป็น จำนวนเต็มเท่านั้น และ เมื่อนำ พารามิเตอร์ x มาแปลงเป็น โครโมโซม จะใช้ ยีน 4 ยีน ในการแทน พารามิเตอร์ x ดังตารางที่ 3-1

Integer	Binary code	Integer	Binary code	Integer	Binary code
1	0001	6	0110	11	1011
2	0010	7	0111	12	1100
3	0011	8	1000	13	1101
4	0100	9	1001	14	1110
5	0101	10	1010	15	1111

ตาราง 3-1 แสดงรูปแบบการแทนโครโมโซมด้วยไบนารีสตริง ขนาด 4 บิต

กำหนดจำนวนประชากรโครโมโซม N เท่ากับ 6 ส่วนความน่าจะเป็นในการ คrossover เท่ากับ 0.7 และความน่าจะเป็นในการ มิวเทชัน เท่ากับ 0.001 [5] (เป็นค่าที่ใช้กันเป็นสากล)

ฟังก์ชันความเหมาะสม คือ $f(x) = 15x - x^2$

เริ่มด้วยการสร้างประชากร โครโมโซมเริ่มแรก จำนวน 6 โครโมโซม โดยการสุ่มเลือกค่า 1 และ 0 เพื่อมาสร้างไบนารีสตริงที่มีความยาวเท่ากับ 4 ต่อมาเป็นการวัดค่าความเหมาะสมให้กับ โครโมโซม แต่ละตัว โดยใช้ฟังก์ชันความเหมาะสมตามตาราง 3-2 จากนั้นทำการปรับปรุง ประชากรโครโมโซม เริ่มต้นด้วยการ เลือก โครโมโซม มาเข้าคู่ เพื่อทำการ มิวเทชัน crossover

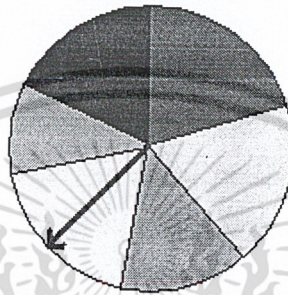
โครโมโซม	รูปแบบโครโมโซม (binary code)	รูปแบบโครโมโซม (Integer)	ค่าความเหมาะสมของโครโมโซม	สัดส่วนของค่าความเหมาะสม (%)
X1	1100	12	36	16.5
X2	0100	4	44	20.2
X3	0001	1	14	6.4
X4	1110	14	14	6.4
X5	0111	7	56	25.7
X6	1001	9	54	24.8

ตาราง 3-2 แสดงประชากรโครโมโซม และค่าความเหมาะสมของโครโมโซมในเจเนอเรชันแรก

คอลัมน์ สุดท้ายในตาราง 3-2 นั้นบ่งบอก อัตราส่วนความเหมาะสม(Fitness ratio) ของ โครโมโซม ตัวนั้นต่ออัตราความเหมาะสมรวมของประชากรโครโมโซมทั้งหมด ซึ่งค่านี้จะเป็นตัวบ่งบอกว่า โครโมโซมตัวนั้นมีโอกาสถูกเลือกมาเข้าคู่มามากน้อยเพียงใด เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางจะเห็นได้ว่า โครโมโซม X5,X6 มีโอกาสที่จะถูกเลือกมากกว่า โครโมโซม X3,X4

วิธีหนึ่งซึ่งมักใช้ในการเลือกโครโมโซม ก็คือการใช้ **roulette wheel selection** [6] (Goldberg,1989 ;Davis,1991) รูปที่ 3-3 เป็นการแสดง roulette wheel ตัวอย่าง ซึ่งจะเห็นได้ว่า โครโมโซม แต่ละตัวจะได้รับเนื้อที่บน roulette wheel แตกต่างกันไป ซึ่งเนื้อที่นั้นจะมีค่าเท่ากับ สัดส่วนความเหมาะสมของโครโมโซมแต่ละตัว โครโมโซมใดมีพื้นที่บน roulette wheel มาก ก็จะมี โอกาสถูกเลือกมากตามไปด้วย



รูป 3-3 แสดง Roulette Wheel ตัวอย่าง

ในการเลือกโครโมโซมเพื่อมาเข้าคู่ นั้น จะทำโดยการสุ่มค่าตั้งแต่ 0 – 100 โดยถ้าเลข นั้นตกอยู่ใน พื้นที่ส่วนของ ของโครโมโซมตัวใด โครโมโซม ตัวนั้นจะเป็นตัวถูกเลือก

จากกรณีศึกษา ที่ 1 จะเห็นได้ว่า ประชากรโครโมโซมเริ่มต้น มีจำนวน 6 โครโมโซม และในการที่จะสร้างโครโมโซม ในเจนเนอเรชันถัดไปนั้น roulette wheel จะต้องทำการหมุนทั้งหมด 6 ครั้ง โดยการเลือก 2 ครั้งแรก X6,X2 อาจจะถูกเลือกมาเพื่อเป็น โครโมโซมต้นแบบ และ โครโมโซม ต้นแบบ คู่ถัดมาอาจเป็น X1 และ X5 ส่วนการเลือก 2 ครั้งสุดท้าย อาจจะได้ X2, X5 โดยหลังจาก ที่โครโมโซมต้นแบบถูกเลือกมาแล้ว จะถูกนำมาครอสโอเวอร์ เพื่อเป็นการปรับปรุงคุณภาพของ โครโมโซม

กรณีศึกษาที่ 2 ปัญหาการหาค่าสูงสุดของฟังก์ชัน $y = x^2$ โดยที่ x มีค่าระหว่างจำนวนเต็ม [0,31]

- การกำหนดฟังก์ชันความเหมาะสม

ให้ ฟังก์ชันเป้าหมาย คือ $f = x^2$

และ ฟังก์ชันความเหมาะสม คือ $F = x^2$

ซึ่งคำตอบที่ดีที่สุด คือ ค่า x ที่มีค่าความเหมาะสมสูงสุด ($\text{Max}(F)$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การกำหนดรูปแบบโครโมโซม

รูปแบบของโครโมโซมสำหรับปัญหานี้จะเป็นแบบไบนารี โดยค่าตัวแปรหรือพารามิเตอร์ของปัญหาจะถูกแปลงให้อยู่ในรูปของไบนารีโครโมโซม คือประกอบด้วยบิตที่มีค่าเป็น 0 หรือ 1 ซึ่งเป็นค่าในเลขฐานสอง และมีความยาวซึ่งเราสามารถกำหนดได้ เช่น ในการเข้ารหัสแบบไบนารีโดยแปลงค่าพารามิเตอร์ x ให้อยู่ในรูปไบนารี 5 บิต ดังนั้นโครโมโซมของปัญหาจะมีค่าอยู่ในช่วง 00000 ถึง 11111 ซึ่งเมื่อถอดรหัสแล้วจะทำให้ x มีค่าอยู่ในช่วง 0 ถึง 32 ตามที่ต้องการ

- การทำงานของเจเนติก อัลกอริทึม

รายละเอียดการทำงานของ เจเนติกอัลกอริทึม นั้นมีขั้นตอนการทำงานพื้นฐานดังต่อไปนี้

(1) ประชากรรุ่นเก่า (Old Population) เป็นชุดของโครโมโซมที่จะถูกคัดเลือกไปเป็นต้นแบบสำหรับสร้างประชากรรุ่นใหม่ (New Population) ในวิวัฒนาการรุ่น (Generation :gen) ต่อไปโดยประชากรเริ่มต้นที่ $gen = 0$ จะถูกสร้างขึ้นโดยการสุ่มตามจำนวนโครโมโซมในแต่ละรุ่น (Chromosome population) ที่กำหนด เช่น ในรุ่นแรกอาจได้ชุดโครโมโซมเป็นดังตาราง 3-3

ลำดับ	โครโมโซม
1	01110
2	11001
3	01000
4	10011

ตาราง 3-3 แสดงประชากรโครโมโซมในเจเนอเรชันแรก

ชุดโครโมโซมในรุ่นแรกนี้ประกอบด้วยโครโมโซม 4 โครโมโซม ซึ่งแต่ละโครโมโซมเกิดจากการสุ่มค่าไบนารี 0 หรือ 1 จำนวน 5 ครั้ง

(2) การวิเคราะห์ค่าความเหมาะสม เป็นขั้นตอนการถอดรหัสจากรูปแบบโครโมโซมที่กำหนดไว้ เพื่อคำนวณค่าความเหมาะสม ตามฟังก์ชันความเหมาะสมของปัญหาสำหรับ แต่ละโครโมโซมในเจเนอเรชัน โดยในกรณีศึกษาที่ ฟังก์ชันเป้าหมายหรือฟังก์ชันความเหมาะสมคือ $F = x^2$ นั่นก็คือการถอดรหัสเลขฐานสองของแต่ละโครโมโซมเป็นตัวแปร x และคำนวณค่าความเหมาะสม คือ x^2

ลำดับ	โครโมโซม	X	ค่าความเหมาะสม
1	01110	14	196
2	11001	25	625
3	01000	8	64
4	10011	19	361

ตาราง 3-4 แสดงค่าความเหมาะสมของโครโมโซมในเจเนอเรชันแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(3) การคัดเลือกโครโมโซม เป็นขั้นตอนที่จำลองแบบการคัดเลือกตามธรรมชาติเพื่อสร้างเมตติงพูล (Mating pool) โดยคัดเลือกโครโมโซมรุ่นเก่าให้เป็น โครโมโซมต้นแบบหรือโครโมโซมพ่อแม่ เพื่อใช้สร้างโครโมโซมลูกในรุ่นต่อไป ซึ่งหลักในการเลือก โครโมโซมพ่อแม่ นั้นจะทำการเลือกตามค่าความเหมาะสมของแต่ละโครโมโซมที่ได้จากฟังก์ชันความเหมาะสม โดยโครโมโซมที่มีค่าความเหมาะสมมากก็จะมีแนวโน้มที่จะถูกเลือกสูงด้วย โดยการกำหนดค่าความน่าจะเป็นที่จะถูกเลือกแต่ละครั้ง (Probability of Selected Value : p_{select}) คำนวณได้จาก ค่าความเหมาะสมของโครโมโซมนั้นเทียบกับผลรวมของค่าความเหมาะสมทั้งหมด ดังสมการที่ 3.1

$$p_{select}_i = \frac{F_i}{\sum F} \quad (3.1)$$

ลำดับ	โครโมโซม	X	ค่าความเหมาะสม (F)	ค่าความน่าจะเป็น(p_{select})
1	01110	14	196	0.157
2	11001	25	625	0.502
3	01000	8	64	0.051
4	10011	19	361	0.290

ตาราง 3-5 แสดงค่าความน่าจะเป็นของแต่ละโครโมโซมที่จะถูกเลือก

จะเห็นได้ว่าการเลือกโครโมโซมต้นแบบหรือโครโมโซมพ่อแม่จาก 4 โครโมโซมนี้โอกาสที่จะสุ่มได้โครโมโซมลำดับที่ 1 ต่อการสุ่มแต่ละครั้งคือ 0.157 และโอกาสที่จะสุ่มได้โครโมโซมลำดับที่ 2,3 และ 4 ต่อการสุ่มในแต่ละครั้งเท่ากับ 0.502 , 0.051 และ 0.290 ตามลำดับ

จากนั้นทำการสุ่มเลือกโครโมโซมต้นแบบโดยใช้วิธี **Roulette Wheel Selection** ซึ่งจะเห็นได้ว่าโครโมโซมลำดับที่ 2 มีค่าความเหมาะสมสูงที่สุดจะมีโอกาสถูกคัดเลือกจำนวนมากที่สุด และโครโมโซมลำดับที่ 3 มีโอกาสถูกเลือกน้อยที่สุด

(4) กระบวนการทางพันธุศาสตร์ เป็นการจำลองแบบทางพันธุกรรม ซึ่งมีตัวดำเนินการ คือ คrossover และ mutation

- crossover เป็นตัวดำเนินการที่ทำหน้าที่ในการแลกเปลี่ยนส่วนของโครโมโซมพ่อแม่ ซึ่งจะดำเนินการตามอัตราของความน่าจะเป็นในการ crossover (Probability of Crossover : P_c) เพื่อสร้างชุดโครโมโซมรุ่นใหม่ หรือ โครโมโซมลูก ซึ่งมีขั้นตอนการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

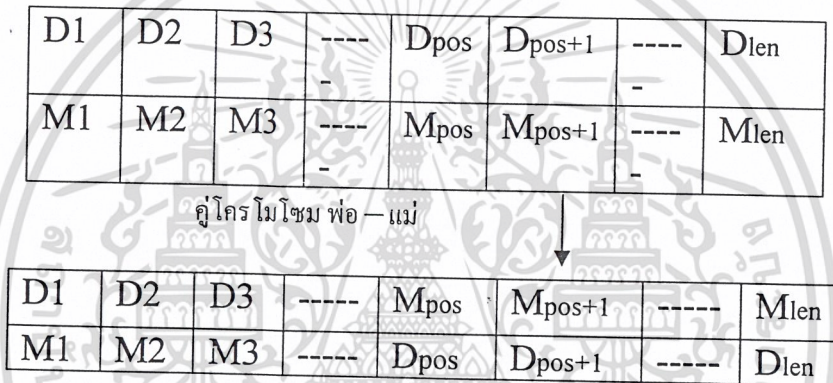
ขั้นตอนแรก สุ่มเลือกโครโมโซมพ่อแม่ขึ้นมา 1 คู่โดยใช้วิธีดั่งที่ได้กล่าวมาแล้ว

ขั้นตอนที่สอง สุ่มเลขจำนวนจริง ซึ่งมีค่าอยู่ในช่วง 0 ถึง 1 โดยถ้าค่าที่สุ่มมามีค่าน้อยกว่าหรือเท่ากับความน่าจะเป็นในการครอสโอเวอร์แล้ว โครโมโซมพ่อแม่จึงจะเกิดการครอสโอเวอร์ขึ้น

ขั้นตอนที่สาม ครอสโอเวอร์โดยการแลกเปลี่ยนส่วนของคู่โครโมโซมพ่อแม่ ซึ่งในกรณีศึกษา นี้ กำหนดให้เป็นการครอสโอเวอร์แบบ 1 จุด (One-point Crossover) มีขั้นตอนดั่งนี้

- สุ่มเลือกตำแหน่ง pos เป็นตำแหน่งที่จะทำการครอสโอเวอร์ ซึ่งค่า pos นี้จะต้องมีค่าตั้งแต่ 1 ถึง ความยาวของโครโมโซม - 1

- แลกเปลี่ยนค่าในแต่ละบิตของคู่โครโมโซมพ่อแม่ตั้งแต่ตำแหน่งที่ pos จนกระทั่งสิ้นสุดโครโมโซม ซึ่งจะทำให้เกิดโครโมโซมลูกตัวใหม่อีก 2 โครโมโซม



จำนวนครั้งในการครอสโอเวอร์ในแต่ละรุ่น(Generation)จะมากน้อยเพียงใดนั้น จะขึ้นอยู่กับกำหนัดค่า Pc ซึ่งเป็นค่าที่มีความแตกต่างกันในแต่ละปัญหา เช่น ถ้าจำนวนประชากรในแต่ละรุ่น (Chromosome population) มีค่าเท่ากับ 30 โครโมโซม และ กำหนดให้ Pc = 0.6 แล้ว จำนวนการครอสโอเวอร์ในแต่ละรุ่นจะเท่ากับ Pc*(population size / 2) = 9 ครั้ง (หาร 2 เนื่องจากการครอสโอเวอร์หนึ่งครั้งเกิดจาก สอง โครโมโซม)

- มิวเตชัน เป็นตัวดำเนินการผ่าเหล่าตัวหนึ่งซึ่งช่วยให้โครโมโซม มีค่าความเหมาะสมที่ ดีขึ้นได้ หลังจากที่ทำกรครอสโอเวอร์แล้ว โดยการกลับบิตเป็นค่าใหม่ในตำแหน่งบิตที่สุ่มได้ ตามอัตราความน่าจะเป็นของการมิวเตชันในแต่ละบิต(Probability of Mutation : Pm) ที่กำหนด ซึ่งสำหรับปัญหานี้ใช้ไบนารีมิวเตชันในการแก้ปัญหา โดยการกลับค่า บิตจาก 0 เป็น 1 หรือ 1 เป็น 0

(5) ประชากรรุ่นใหม่ เป็นชุดโครโมโซมลูกที่เกิดจากขั้นตอนของวิวัฒนาการต่างๆทั้งหมด ซึ่ง ประชากรรุ่นใหม่ที่เกิดขึ้นจะเป็นประชากรรุ่นเก่าสำหรับวิวัฒนาการรุ่นต่อไป ซึ่งเรียกรวมการแบบนี้ ว่า การรีโพรดักชันแบบทั่วไป (General Reproduction) และจะปฏิบัติกระบวนการนี้ซ้ำๆ จนกระทั่ง ถึงรุ่นที่มากที่สุด(Max generation : maxgen)

บทที่ 4

การออกแบบ เจเนติกอัลกอริทึมเพื่อแก้ปัญหา

ในบทนี้จะเป็นการออกแบบเจเนติกอัลกอริทึมซึ่งเหมาะกับการแก้ปัญหาการบริหารพอร์ตการลงทุน ซึ่งจะกล่าวถึงตั้งแต่วิธีการแทนโครโมโซมที่เหมาะสม การครอสโอเวอร์ และการมิวเตชันที่เหมาะสมสำหรับนำมาประยุกต์ใช้กับปัญหานี้

4.1 รูปแบบการแทนโครโมโซม

โดยทั่วไปแล้วการแทนโครโมโซมในเจเนติกอัลกอริทึมนั้นมักเป็นการแทนโครโมโซมในลักษณะไบนารี (Binary Representation) แต่ในปัญหาการบริหารพอร์ตการลงทุนนั้น โครโมโซม 1 โครโมโซมจะประกอบด้วยยีนซึ่งหมายถึงจำนวนหน่วยของหุ้นกู้แต่ละตัวในพอร์ตการลงทุน 1 พอร์ตคั้งนั้นจึงไม่สามารถแทนโครโมโซมอยู่ในรูปแบบของไบนารีได้ เนื่องจากยีนซึ่งถูกแทนด้วยไบนารีนั้นใช้แทนจำนวนหน่วยของหุ้นกู้ได้ไม่สะดวกนัก

ในปัญหานี้จึงถูกออกแบบให้แทนโครโมโซมด้วยอาร์เรย์ของจำนวนจริงที่เป็นบวก โดยสมาชิกแต่ละตัวของอาร์เรย์นั้นแทนจำนวนหน่วยของหุ้นกู้ในพอร์ต (สมาชิกแต่ละตัวของอาร์เรย์แทนยีนหนึ่งยีน) การแทนโครโมโซมในลักษณะนี้จะคิดหน่วยของทั้งพอร์ตเป็น 100% นั่นคือ 1 ตัวอย่างการแทนโครโมโซมในลักษณะนี้จะเป็น ดังรูปที่ 4-1 เช่น 0.05 ซึ่งเป็นยีนตัวแรกของโครโมโซมจะเป็นหน่วยของหุ้นกู้ที่ชื่อ LB04NA หมายความว่า LB04NA เป็น 5% ของทั้งพอร์ต

LB04NA	AIS070A	SHIN075A	TRUE087A	TAC090A
(0.05	0.2	0.2	0.4	0.15)

รูป 4-1 ตัวอย่างโครโมโซม

ในตอนเริ่มของกระบวนการเจเนติกอัลกอริทึมในแอปพลิเคชันนี้จะทำการสร้าง combination ของหุ้นกู้ที่เป็นไปได้ทั้งหมดจากเซตของหุ้นกู้ที่ผู้ใช้ได้ทำการอัปโหลดขึ้นมา โดยมียีนในโครโมโซมเป็นจำนวนเท่ากับจำนวนของหุ้นกู้ที่ผู้ใช้ต้องการให้แอปพลิเคชันทำการเลือกให้ เช่น จากรูปที่ 4-1 เป็น คอมบินชันของพอร์ตที่ผู้ใช้ต้องการให้มีหุ้นกู้ 5 ตัวด้วยกัน ซึ่งจำนวนหน่วยของหุ้นกู้ที่จะนำมาแทนในยีนแต่ละตัวนั้น เริ่มแรกจะได้มาจากการสุ่มโดยสุ่มจากตัวเลขในช่วง ค่ามากกว่า 0 แต่น้อยกว่า 1

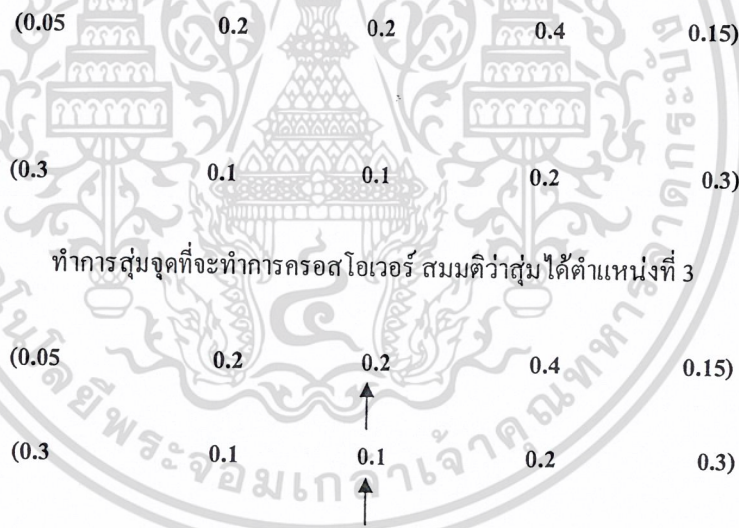
4.2 การครอสโอเวอร์ (Crossover)

การครอสโอเวอร์นั้นมีอยู่หลากหลายรูปแบบด้วยกันและในการครอสโอเวอร์แต่ละแบบนี้จะให้ผลแตกต่างกันออกไป ในส่วนของปัญหานี้การครอสโอเวอร์บางแบบก็ให้โครโมโซมลูกซึ่งเป็นโครโมโซมที่ไม่ถูกต้องหรือเป็นวิธีที่ยู่ยากเกินไป ดังนั้นจึงต้องเลือกวิธีการครอสโอเวอร์ที่เหมาะสมกับปัญหานี้ด้วย โดยในที่นี้จะกล่าวถึงเฉพาะการครอสโอเวอร์ที่เหมาะสมที่จะนำมาประยุกต์ใช้กับปัญหานี้เท่านั้น

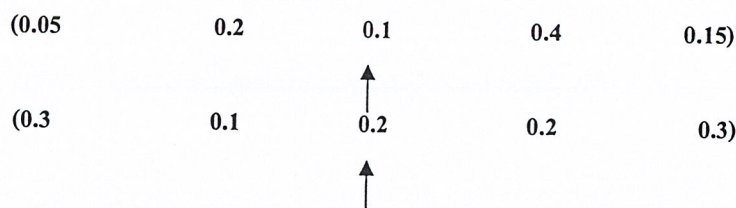
4.2.1 การครอสโอเวอร์แบบดั้งเดิม (Classical Crossover)

การครอสโอเวอร์แบบดั้งเดิมถูกนำเสนอโดย Holland ในปี ค.ศ. 1975 ซึ่งขั้นตอนการทำงานของครอสโอเวอร์แบบดั้งเดิมนั้นจะมีการนำมาปรับปรุงเล็กน้อยในแอปพลิเคชันนี้ โดยจะไม่ทำการครอสโอเวอร์ตั้งแต่ตำแหน่งที่สุ่มได้ไปจนกระทั่งสิ้นสุดโครโมโซมแต่จะทำการครอสโอเวอร์เฉพาะยีนในจุดที่สุ่มได้เท่านั้น ซึ่งสามารถพิจารณาได้จากตัวอย่างต่อไปนี้

และ



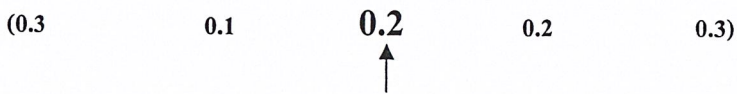
จากนั้นทำการครอสโอเวอร์จะได้โครโมโซมใหม่เป็น



รูป 4-2 แสดงการครอสโอเวอร์

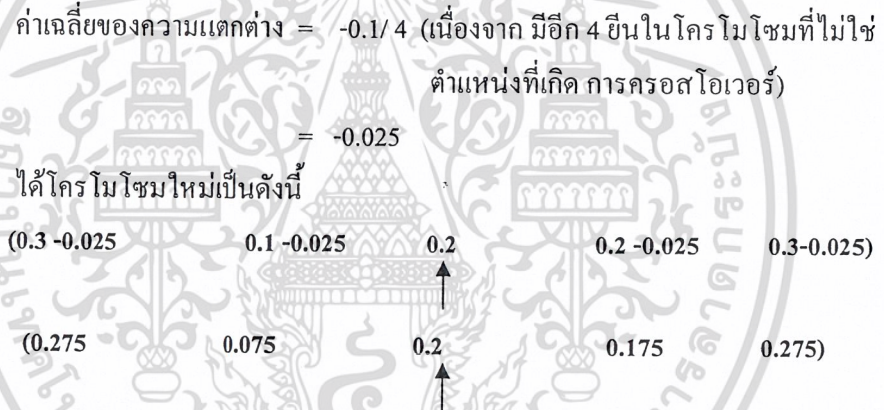
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจะเห็นว่าจากการครอสโอเวอร์แบบนี้ จะทำให้ยีนในโครโมโซมซึ่งก็คือจำนวนหน่วยของหุ่นกู่แต่ละตัวในแต่ละโครโมโซมมีการเปลี่ยนแปลงไปเรื่อยๆในแต่ละเจเนอเรชันนั่นเอง โดยการครอสโอเวอร์นี้จะเกิดขึ้นตามความน่าจะเป็นในการครอสโอเวอร์ แต่จากการครอสโอเวอร์ก็ทำให้หน่วยรวมของทั้งพอร์ตไม่เท่ากับ 1 (100%) ดังนั้นจึงต้องมีการ Reweight หน่วยของหุ่นกู่ภายในโครโมโซมใหม่ทุกครั้งโดยวิธีที่จะเลือกใช้ก็คือ การเฉลี่ยค่า เช่น



รูป 4-3 แสดงโครโมโซมก่อนทำการครอสโอเวอร์

การ reweight ค่ายีนในโครโมโซมจะทำโดยการนำค่าความแตกต่างระหว่างค่าหน่วยที่ยีนตำแหน่งนั้นก่อนการครอสโอเวอร์และค่าหน่วยที่ยีนตำแหน่งนั้นหลังการครอสโอเวอร์ ซึ่งจากตัวอย่างก็คือ -0.1 ($0.1 - 0.2$) มาเฉลี่ยให้กับค่าหน่วยในตำแหน่งอื่นๆของโครโมโซม ยกเว้นค่าหน่วยในตำแหน่งที่ถูกครอสโอเวอร์ ดังนี้



จะเห็นได้ว่าหลังจากทำการ reweight แล้วค่ารวมของหน่วยของทั้งโครโมโซมจะมีค่าเท่ากับ 1 เหมือนเดิม

4.3 การมิวเตชัน(Mutation)

การมิวเตชันนั้น เป็นตัวดำเนินการทางพันธุศาสตร์อย่างหนึ่งซึ่งจะทำการผ่าเหล่าที่ทำให้โครโมโซมมีความเหมาะสมที่ดีขึ้น โดยการมิวเตชันนั้นมีอยู่หลากหลายรูปแบบ แต่ละแบบก็มีความเหมาะสมกับปัญหารูปแบบที่แตกต่างกันออกไป ในส่วนของปัญหานี้ ไม่สามารถใช้การมิวเตชันแบบดั้งเดิมได้ พิจารณาได้จากรูปที่ 4-4



รูป4-4 แสดงการมิวเตชันแบบดั้งเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4-4 เป็นการมิวเตชันแบบดั้งเดิม (Classical Mutation) ที่เริ่มต้นด้วยการสุ่มตำแหน่งยีนที่จะทำการมิวเตชัน ซึ่งจากรูปทำการสุ่มได้ตำแหน่งที่ 4 แล้วทำการกลับบิตในตำแหน่งที่สุ่มได้นั้นเป็นค่าใหม่ตามอัตราความน่าจะเป็นในการเกิดมิวเตชัน โดยการมิวเตชันลักษณะดังกล่าวมักจะกับปัญหาที่มีการแทนโครโมโซมแบบไบนารีสตริงเท่านั้น สำหรับการแทนโครโมโซมโดยให้ยีนแต่ละยีนถูกแทนจำนวนจริงที่มากกว่าศูนย์อย่างที่ได้กล่าวมาแล้วข้างต้นนั้นจำเป็นต้องใช้วิธีการมิวเตชันรูปแบบอื่น โดยสามารถพิจารณาได้ดังรูปที่ 4-5

(0.05	0.2	0.1	0.4	0.15)
		↓		
(0.05	0.2	0.2	0.4	0.15)

รูป 4-5 แสดงการมิวเตชันแบบสุ่มเลือก

จากรูปที่ 4-5 จะทำการมิวเตชันแบบสุ่มเลือกโดยเริ่มจากการสุ่มตำแหน่งที่จะทำการมิวเตชัน ซึ่งจากรูปทำการสุ่มได้ตำแหน่งที่ 3 แต่ในการมิวเตชันรูปแบบนี้จะทำโดยการสุ่มยีนใหม่ (จำนวนจริงที่อยู่ในช่วงที่เหมาะสม ตัวใหม่ นั่นคือ 0-1) จะมาแทนที่ในตำแหน่งที่ต้องการทำมิวเตชัน โดยการทำการมิวเตชันในรูปแบบนี้ยังคงต้องดูค่าความน่าจะเป็นในการเกิดมิวเตชันเช่นกัน ซึ่งหลังจากมีการมิวเตชันแล้วก็ต้องทำการ reweight คำนวณที่ได้กล่าวมาแล้วทุกครั้ง

บทที่ 5

การปรับปรุงประสิทธิภาพของ เจเนติกอัลกอริทึม

การประยุกต์ใช้ เจเนติกอัลกอริทึม ในการหาคำตอบของปัญหาต่างๆ นั้น ประสิทธิภาพในการค้นหาคำตอบของ เจเนติกอัลกอริทึม นั้นค่อนข้างที่จะแตกต่างกัน หากปัญหาใดมีข้อกำหนด หรือเงื่อนไขที่ซับซ้อนมากแล้ว การดำเนินการของ เจเนติกอัลกอริทึม ก็จะยุ่งยาก และใช้เวลาค่อนข้างมากในการหาคำตอบ ซึ่งเจเนติกอัลกอริทึมอาจไม่สามารถที่จะหาคำตอบที่ดีที่สุดได้ภายในระยะเวลาที่กำหนดได้ ดังนั้นจึงต้องมีการปรับปรุงประสิทธิภาพของ เจเนติกอัลกอริทึม ในส่วนของการคัดเลือกซึ่งประกอบด้วยรูปแบบการคัดเลือกต้นแบบในการสร้าง เมตติงพูล (mating pool) และวิธีในการสุ่มเลือกโครโมโซมต้นแบบ รวมถึงในส่วนของการรีโปรดัคชันด้วยการรักษาโครโมโซมที่ดีแบบต่างๆ

5.1 รูปแบบของการคัดเลือกต้นแบบ

การหาคำตอบของปัญหาต่างๆ โดยทั่วไปนั้นจะแบ่งได้เป็น 2 แบบ คือ การหาคำตอบของปัญหาที่ให้ประโยชน์สูงสุด (Maximize Utility Function) และการหาคำตอบของปัญหาที่ใช้ค่าใช้จ่ายต่ำที่สุด (Minimize Cost Function) ซึ่งหากประยุกต์ใช้ เจเนติกอัลกอริทึม ช่วยในการหาคำตอบนั้นจะเป็นการหาคำตอบที่ต้องการค่าฟังก์ชันเป้าหมายมากที่สุด (Maximization Objection Function : MAX (f)) หรือเป็นการหาคำตอบที่ให้ค่าฟังก์ชันเป้าหมายน้อยที่สุด (Minimization Objection Function : MIN (f)) โดยทั่วไปมักจะกำหนดฟังก์ชันเป้าหมายของปัญหาให้เป็นฟังก์ชันเดียวกับฟังก์ชันความเหมาะสมของแต่ละ โครโมโซมของ เจเนติกอัลกอริทึม นั้น โครโมโซมที่ดี (best) ควรที่จะมีค่าของความเหมาะสมที่สูงที่สุด ส่วนโครโมโซมที่ไม่ดีนั้น (worst) ควรจะมีค่าของความเหมาะสมที่ต่ำที่สุด เพื่อที่จะใช้เป็นตัวตัดสินใจในการเลือกโครโมโซมต้นแบบของเมตติงพูล ซึ่งเป็นลักษณะเดียวกับคำตอบของการหาค่าสูงสุดของปัญหา แต่สำหรับการหาค่าต่ำที่สุดของปัญหาจะใช้ฟังก์ชันเป้าหมายที่ให้ค่าต่ำที่สุด ซึ่งโครโมโซมที่ดีนั้นคือโครโมโซมที่มีค่าข้อมูลดิบที่ต่ำทำให้มีค่าของความเหมาะสมต่ำ และโครโมโซมที่ไม่ดีจะมีค่าของข้อมูลดิบสูงทำให้มีค่าความเหมาะสมสูง ดังนั้น จึงต้องมีการปรับปรุงฟังก์ชันเป้าหมายให้เป็นแบบที่ถูกต้องสำหรับการวัดประสิทธิภาพของแต่ละโครโมโซมของ เจเนติกอัลกอริทึม ซึ่งวิธีการปรับปรุงฟังก์ชันเป้าหมายให้เป็นฟังก์ชันความเหมาะสมสำหรับการหาค่าสูงสุด และต่ำสุดของปัญหา แสดงได้ดังสมการที่ 5.1

$$F = \begin{cases} f \text{ สำหรับ MAX (f)} \\ 1/(1+f) \text{ สำหรับ MIN (f)} \end{cases} \quad (5.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าความเหมาะสมของแต่ละโครโมโซมจะถูกพิจารณาในการคัดเลือกในการสร้างเมตคิงพูล ซึ่งวิธีการคัดเลือกต้นแบบโดยทั่วไปของ เจเนติกอัลกอริทึม นั้นมีด้วยกัน 2 รูปแบบคือ

5.1.1 การคัดเลือกแบบอ้างอิงค่าความเหมาะสม (fitness-based) เป็นรูปแบบของการคัดเลือกโครโมโซมต้นแบบโดยการวัดประสิทธิภาพของแต่ละโครโมโซมจากค่าของความเหมาะสมโดยตรง คือโครโมโซมที่มีค่าความเหมาะสมสูง จะมีโอกาสถูกเลือกเป็นต้นแบบมากกว่าโครโมโซมที่มีความเหมาะสมต่ำ และเนื่องจากการดำเนินการของ เจเนติกอัลกอริทึม นั้น โครโมโซมในรุ่นแรกๆ ส่วนใหญ่จะเป็นโครโมโซมที่ไม่ดีพอ แต่ในบางครั้งก็อาจมีโครโมโซมที่ดีเกินไป (Superchromosome) เกิดขึ้นซึ่งทำให้ค่าความคาดหวังที่จะสุ่มได้ของโครโมโซมนั้นสูงกว่าโครโมโซมอื่นๆอยู่มาก และทำให้โครโมโซมนั้นถูกสุ่มเป็นต้นแบบในจำนวนครั้งที่มากเกินไป อันเป็นสาเหตุให้เกิดจุดสิ้นสุดก่อนที่ควร (Premature Convergence) คือ เจเนติกอัลกอริทึม จะหาคำตอบได้ก่อนซึ่งไม่ใช่คำตอบที่ดีที่สุด หรือในกรณีของการดำเนินการของ เจเนติกอัลกอริทึม ในรุ่นหลังๆ นั้นโครโมโซมเริ่มที่จะดีขึ้นแล้วทำให้ความแตกต่างของแต่ละโครโมโซมนั้นน้อยลงมาก จนทำให้ค่าของความคาดหวังที่จะสุ่มได้ของแต่ละโครโมโซมใกล้เคียงกันมาก ซึ่งหมายความว่าจำนวนโครโมโซมต้นแบบของแต่ละโครโมโซมก็จะมีค่าใกล้เคียงกัน มีผลทำให้ความสามารถในการค้นหาคำตอบของ เจเนติกอัลกอริทึม นั้นลดลงจนเหมือนกับว่าการหาคำตอบนั้นเป็นการหาคำตอบแบบสุ่ม ดังนั้นควรทำการปรับปรุงฟังก์ชันความเหมาะสมโดยการปรับสัดส่วนค่าของความเหมาะสม

เพื่อปรับค่าของความคาดหวังที่จะสุ่มได้ของแต่ละโครโมโซมให้มีความเหมาะสมมากยิ่งขึ้น โดยทั่วไปนั้นจะมีอยู่ 3 วิธี คือ

5.1.1.1 แบบหน้าต่าง (Windowing) เป็นวิธีการปรับค่าความเหมาะสมของแต่ละโครโมโซมโดยพิจารณาจากค่าของความเหมาะสมที่ไม่ดี ดังสมการที่ 5.2

$$F = \begin{cases} f - f_{\text{worst}} & \text{สำหรับ MAX (f)} \\ f_{\text{worst}} - f & \text{สำหรับ MIN (f)} \end{cases} \quad (5.2)$$

เช่น ในการหาคำตอบที่ต้องการค่าสูงสุดของปัญหา ซึ่งประกอบด้วยโครโมโซมที่มีค่าของความเหมาะสมคือ 105,115,110,140 โดยกำหนด $f_{\text{worst}} = 100$ แล้วคำนวณค่า p_{select_i} และค่า E_i

ลำดับ โครโมโซม	ไม่ปรับสัดส่วน			ปรับสัดส่วน		
	ค่าความ เหมาะสม ($F=f$)	ค่าความ น่าจะเป็น (p_{select_i})	จำนวนที่ คาดหวัง (E_i)	ค่าความ เหมาะสม ($F=f-f_{worst}$)	ค่าความ น่าจะเป็น (p_{select_i})	จำนวนที่ คาดหวัง (E_i)
1	105	0.2234	0.8936	$105-100=5$	0.0714	0.2857
2	115	0.2447	0.9787	$115-100=15$	0.2143	0.8571
3	110	0.2340	0.9362	$110-100=10$	0.1429	0.5714
4	140	0.2979	1.1915	$140-100=40$	0.5714	2.2857
รวม	470			70		

ตาราง 5-1 แสดงการเปรียบเทียบค่าความเหมาะสมจากการปรับสัดส่วนและไม่ปรับสัดส่วนกับปัญหาแบบหาค่าใช้จ่ายมากที่สุด

จะเห็นได้ว่าค่าของความเหมาะสมของแต่ละโครโมโซม ทำให้มีค่าที่คาดหวังว่าจะสุ่มได้เป็น 0.8936, 0.9787, 0.9362 และ 1.1915 ซึ่งมีค่าใกล้เคียงกันมาก คือ จะทำให้จำนวนต้นแบบในเมตดิงพลของแต่ละโครโมโซมมีประมาณ 1 โครโมโซมเท่านั้น แต่ถ้ามีการปรับสัดส่วนโดยการกำหนดให้ $f_{worst} = 100$ แล้วค่าที่คาดหวังว่าจะสุ่มได้ของแต่ละโครโมโซมจะมีค่าเป็น 0.2857, 0.8571, 0.5714 และ 2.2857 โดยโครโมโซมลำดับที่ 4 มีจำนวนที่คาดหวังสูงขึ้นไป 2 เท่าจากเดิม ทำให้โครโมโซมลำดับที่ 4 มีโอกาสถูกคัดเลือกเป็นต้นแบบมากยิ่งขึ้น ในทำนองเดียวกันการหาค่าต่ำที่สุดของปัญหาซึ่งประกอบด้วยโครโมโซมที่มีค่าความเหมาะสมเป็น 100, 135, 125, 140 โดยกำหนดให้ $f_{worst} = 150$ แล้วเราสามารถเปรียบเทียบการปรับสัดส่วนค่าความเหมาะสม และการไม่ทำการปรับสัดส่วนค่าความเหมาะสมได้ดังนี้

ลำดับ โครโมโซม	ไม่ปรับสัดส่วน				ปรับสัดส่วน		
	ค่าความ เหมาะสม ($F=f$)	ค่าความ เหมาะสม ($F=1/(1+f)$)	ค่าความ น่าจะเป็น เป็น (p_{select_i})	จำนวน ที่ คาดหวัง (E_i)	ค่าความ เหมาะสม ($F=f_{worst}-f$)	ค่าความ น่าจะเป็น (p_{select_i})	จำนวนที่ คาดหวัง (E_i)
1	100	0.0099	0.03067	1.2268	$150-100=50$	0.5000	2.0000
2	135	0.0074	0.2278	0.9111	$150-135=15$	0.1500	0.6000
3	125	0.0079	0.2458	0.9834	$150-125=25$	0.2500	1.0000
4	140	0.0071	0.2197	0.8788	$150-140=10$	0.1000	0.4000
รวม	500	0.0323			100		

ตาราง 5-2 แสดงการเปรียบเทียบค่าความเหมาะสมจากการปรับสัดส่วนและไม่ปรับสัดส่วนกับปัญหาแบบหาค่าใช้จ่ายน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าความเหมาะสมของแต่ละโครโมโซมทำให้จำนวนที่คาดหวังว่าจะสุ่มได้เป็น 1.2268, 0.9111, 0.9834 และ 0.8786 ซึ่งหากปรับสัดส่วนโดยการกำหนดให้ $f_{\text{worst}} = 150$ แล้วค่าของความเหมาะสมจะมีค่าเป็น 50, 15, 25 และ 10 ทำให้ปรับจำนวนที่คาดหวังว่าจะสุ่มได้เป็น 2.0, 0.6, 1.0 และ 0.4 โดยโครโมโซมลำดับที่ 1 มีจำนวนที่คาดหวังสูงขึ้นและสามารถเป็นต้นแบบได้เพียงพอ

โดยทั่วไปนั้นมักใช้ค่าที่ต่ำที่สุดในการหาค่าสูงสุด หรือใช้ค่าสูงที่สุดในการหาค่าต่ำที่สุดของปัญหาเป็นค่า f_{worst} แต่ในการค้นหาคำตอบของ เจเนติกอัลกอริทึม นั้นจะไม่สามารถระบุค่าที่ต่ำที่สุดของปัญหาได้ ดังนั้นวิธีการกำหนด f_{worst} ของ เจเนติกอัลกอริทึม โดยกำหนดให้

$f_{\text{worst}} =$ ค่าที่ไม่ดีที่สุดของโครโมโซมทั้งหมดในรุ่นนั้น

$w =$ ขนาดของหน้าต่าง (window size) หรือจำนวนรุ่นที่ผ่านมาที่ต้องการ

$f_{\text{worst}} = f_{\text{worst}}$ ในรุ่นเริ่มต้น และจะเปลี่ยนแปลงเมื่อเกิดค่า f_{worst} ที่ดีขึ้นถ้า $w = 0$

f_{worst} ของ w รุ่นที่ผ่านมา และจะเปลี่ยนแปลงเมื่อเกิดค่า f_{worst} ที่ดีขึ้นถ้า $w > 0$

การกำหนดขนาดของหน้าต่างนั้นแตกต่างกันไปในแต่ละปัญหา โดยส่วนมากแล้วขนาดของหน้าต่างจะอยู่ในช่วง 1 ถึง 10 รุ่นที่ผ่านมา ($1 \leq w \leq 10$)

5.1.1.2 แบบเชิงเส้น (Linear) เป็นวิธีการปรับค่าความเหมาะสมโดยการกำหนดให้โครโมโซมที่มีค่าความเหมาะสมแบบเฉลี่ย จะต้องมีการปรับค่าความเหมาะสมโดยกำหนดให้โครโมโซมที่คาดหวังว่าจะสุ่มได้ของข้อมูลดิบเฉลี่ย และควบคุมค่าความคาดหวังที่จะสุ่มได้โครโมโซมที่ดีที่สุดเป็นจำนวนเท่าของโครโมโซมที่มีความเหมาะสมเฉลี่ย แล้วทำการปรับค่าความเหมาะสมของโครโมโซมอื่นๆ ในลักษณะเชิงเส้นดังสมการที่ 5.3

$$F = af + b \quad (5.3)$$

โดยที่ $F_{\text{best}} = C_{\text{mult}} * f_{\text{avg}}$

f_{avg} คือค่าของข้อมูลเฉลี่ย, f_{best} คือค่าของข้อมูลดิบที่ดีที่สุด

F_{avg} คือค่าความเหมาะสมเฉลี่ย, F_{best} คือความเหมาะสมที่ดีที่สุด และ

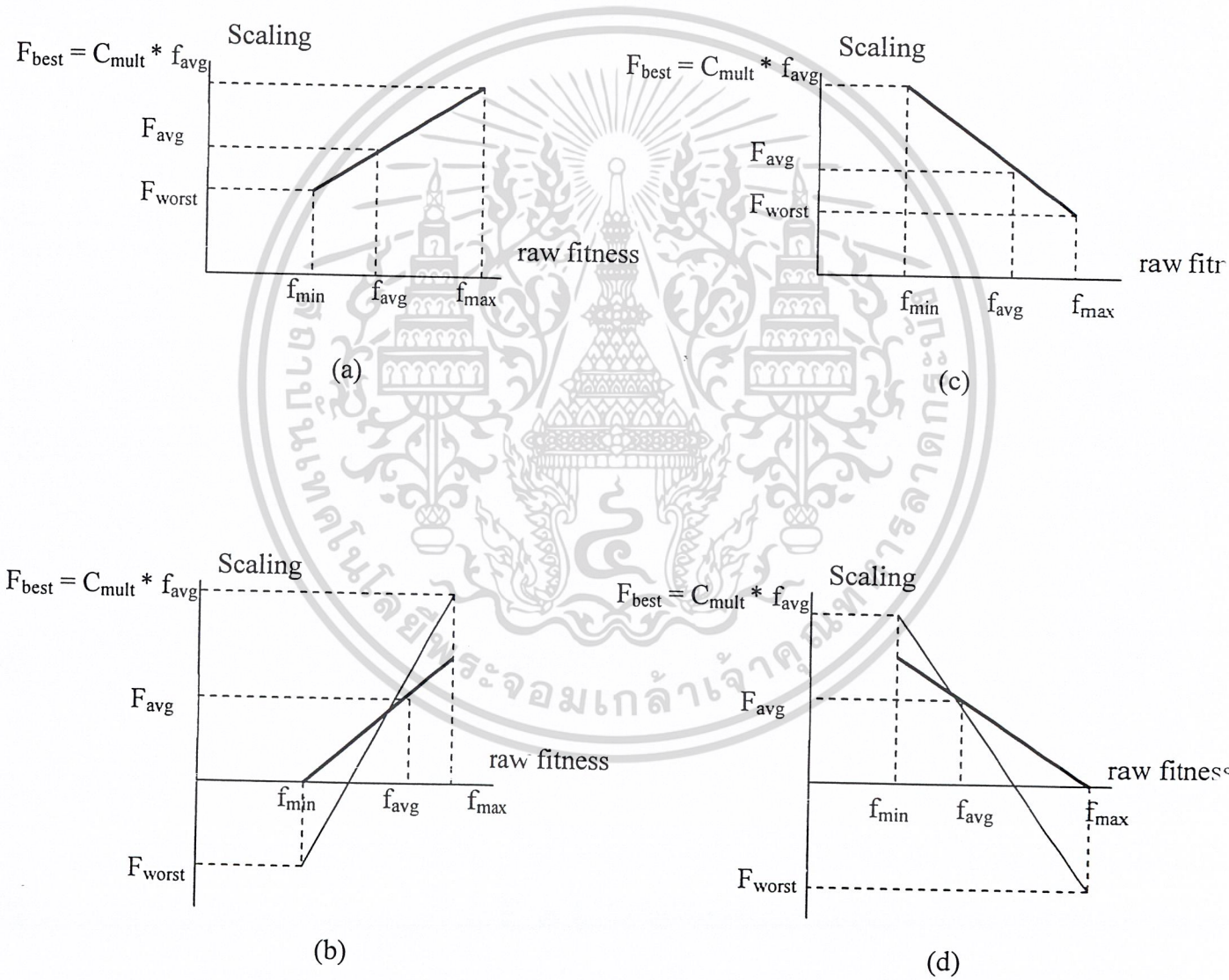
C_{mult} คือจำนวนเท่าของโครโมโซมที่มีค่าความเหมาะสมเฉลี่ย

การปรับสัดส่วนแบบเชิงเส้นของ เจเนติกอัลกอริทึม สำหรับการหาค่าสูงสุดของปัญหานั้น จะเห็นได้ว่าการดำเนินการในรุ่นแรกๆ ถ้าเกิดโครโมโซมที่ดีเกินไปแล้ว จะถูกปรับสัดส่วนให้มีค่าความเหมาะสมลดลง และโครโมโซมที่ไม่ดีจะถูกปรับค่าความเหมาะสมให้สูงขึ้น ดังรูปที่ 5-1a ซึ่งทำให้ไม่เกิดจำนวนที่คาดหวังว่าจะสุ่มได้แต่ละโครโมโซมไม่แตกต่างกันมากเกินไป ซึ่งค่า a และ b กำหนดดังสมการ 5.4 และ 5.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a = \frac{(c_{mult} - 1) * f_{avg}}{f_{best} - f_{avg}} \tag{5.4}$$

$$b = \frac{f_{avg} * (f_{best} - (c_{mult} * f_{avg}))}{f_{best} - f_{avg}} \tag{5.5}$$



รูป 5-1 การปรับสัดส่วนค่าความเหมาะสมแบบเชิงเส้น

เช่นการหาค่าสูงสุดของปัญหาซึ่งประกอบด้วยโครโมโซมที่มีความเหมาะสมคือ 500, 15, 20 และ 90 ซึ่งโครโมโซมลำดับที่ 1 ที่มีค่าความเหมาะสมสูงมากกว่าโครโมโซมอื่น และมีจำนวนที่คาดหวังว่าจะคู่กันได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น 3.2 ซึ่งทำให้โอกาสที่จะสุ่มได้โครโมโซมอื่นน้อยมาก ดังนั้นหากปรับสัดส่วนแบบเชิงเส้น โดยกำหนดให้จำนวนที่คาดหวังว่าจะสุ่มได้เป็นสองเท่าของค่าความเหมาะสมเฉลี่ย ($C_{mult} = 2.0$) แล้วปรับสัดส่วนแบบเชิงเส้นดังนี้

ลำดับ โครโมโซม	ไม่ปรับสัดส่วน			ปรับสัดส่วน		
	ค่าความ เหมาะสม ($F=f$)	ค่าความ น่าจะเป็น (p_{select_i})	จำนวนที่ คาดหวัง (E_i)	ค่าความ เหมาะสม ($F=af+b$)	ค่าความ น่าจะเป็น (p_{select_i})	จำนวนที่ คาดหวัง (E_i)
1	500	0.8000	3.2000	312.5000	0.5000	2.0000
2	15	0.0240	0.0960	92.0455	0.1473	0.5891
3	20	0.0320	0.1280	94.3182	0.1509	0.6036
4	90	0.1440	0.5760	126.1364	0.2018	0.8073
รวม	625					

$$f_{avg} = 156.25$$

$$f_{best} = 500$$

$$a = 0.45$$

$$b = 85.23$$

ตาราง 5-3 แสดงค่าความเหมาะสมโดยใช้วิธีการปรับค่าแบบเชิงเส้น

จะเห็นว่าโครโมโซมลำดับที่ 1 ถูกปรับค่าความเหมาะสมให้ลดลงเป็น 312.5 มีจำนวนที่คาดหวังว่าจะสุ่มได้ลดลงเป็น 2.0 เป็นการปรับจำนวนที่คาดหวังว่าจะสุ่มได้ของโครโมโซมอื่นให้มีโอกาสที่จะถูกสุ่มได้มากขึ้นและ เมื่อผ่านกระบวนการ เจเนติกอัลกอริทึม ไปหลายๆ ครั้ง โครโมโซมในรุ่นหลังนั้นจะมีค่าความเหมาะสมที่ใกล้เคียงกัน แต่อาจมีบางโครโมโซมที่ไม่ดีซึ่งมีความแตกต่างจากโครโมโซมส่วนใหญ่มาก การปรับสัดส่วนนั้น อาจให้ค่าความเหมาะสมที่มีค่าติดลบได้ ดังรูปที่ 5-1b จึงต้องมีการทำการแก้ไขโดยการปรับให้ $f_{min} = 0$ ก่อนจึงค่อยทำการปรับสัดส่วนค่าความเหมาะสมของโครโมโซมอื่นๆ ในลักษณะเชิงเส้นต่อไปซึ่งค่า a' และ b' คำนวณดังสมการ 5.6 และ 5.7

$$a' = \frac{f_{avg}}{f_{avg} - f_{min}} \quad (5.6)$$

$$b' = \frac{-f_{min} * f_{avg}}{f_{avg} - f_{min}} \quad (5.7)$$

เช่นการหาค่าสูงสุดของปัญหาในรุ่นต่างๆ ซึ่งประกอบด้วยโครโมโซมที่มีค่าความเหมาะสมเป็น 900, 240, 575 และ 570 ซึ่งโครโมโซมลำดับที่ 2 มีค่าความเหมาะสมต่ำกว่าโครโมโซมอื่นเกินไปซึ่งต้องทำการปรับสัดส่วนเชิงเส้น โดยกำหนดให้ $C_{mult} = 2.0$ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับ	ไม่ปรับสัดส่วน			ปรับสัดส่วน				
	ค่าความเหมาะสม (F=f)	ค่าความน่าจะเป็น (pselect _i)	จำนวนที่คาดหวัง (E _i)	ค่าความเหมาะสม (F=af+b)	ค่าความน่าจะเป็น (pselect _i)	ค่าความเหมาะสม (F=a'f+b')	ค่าความน่าจะเป็น (pselect _i)	จำนวนที่คาดหวัง (E _i)
1	900	0.3939	1.5755	1142.5000	0.5000	1138.1887	0.4981	1.9925
2	240	0.1050	0.4201	-4.3441	-0.0019	0.0000	0.0000	0.0000
3	575	0.2516	1.0066	577.7662	0.2529	577.7170	0.2528	1.0113
4	570	0.2495	0.9978	569.0779	0.2490	569.0943	0.2491	0.9962
รวม	2285							

$$f_{avg} = 571.25$$

$$f_{best} = 900.00$$

$$f_{min} = 240.00$$

$$a = 1.74 \quad a' = 1.72$$

$$b = -421.38 \quad b' = -413.89$$

ตาราง 5-4 แสดงค่าความเหมาะสมเฉลี่ยโดยใช้วิธีการปรับค่าแบบเชิงเส้น โดยมีโครโมโซมตัวหนึ่งมีค่าความเหมาะสมน้อยเกินไป

จะเห็นว่าเมื่อได้ทำการปรับสัดส่วนแล้วโครโมโซมลำดับที่ 2 จะมีค่าความเหมาะสมติดลบ จึงต้องทำการปรับค่าความเหมาะสมเป็น 0 และปรับสัดส่วนโครโมโซมอื่นแบบเชิงเส้นต่อไป สำหรับการหาค่าต่ำที่สุดของปัญหานั้นจะเป็นลักษณะเดียวกันดังรูปที่ 5-1c และ 5-1d ตามลำดับซึ่งค่าคงที่ของสัมประสิทธิ์ a, b จะถูกคำนวณในการดำเนินการในแต่ละรุ่น และค่า C_{mult} เป็นจำนวนเท่าของค่าความเหมาะสมเฉลี่ยที่ต้องการ โดยทั่วไปควรมีค่าอยู่ในช่วง [1.2, 2.0] ซึ่งจะขึ้นอยู่กับการประยุกต์ใช้ในแต่ละปัญหา

5.1.1.3 แบบสัดส่วนเบี่ยงเบนมาตรฐาน (Sigma Truncation) เนื่องจากวิธีการปรับสัดส่วนแบบเชิงเส้นนั้นอาจจะทำให้เกิดค่าความเหมาะสมที่เป็นค่าลบในรุ่นหลังๆ ของการดำเนินงาน ดังนั้นสามารถแก้ไขได้โดยตัดโครโมโซมที่มีความแตกต่างจากโครโมโซมส่วนใหญ่ทิ้งไป ซึ่งจะใช้ค่าส่วนเบี่ยงเบนมาตรฐาน (σ) ในการพิจารณาตัดโครโมโซมที่มีความแตกต่างนั้นทิ้งไปดังสมการที่ 5.8

$$F = \begin{cases} f - (f_{avg} - c\sigma) & \text{สำหรับ MAX (f) ถ้า } F < 0 \text{ กำหนดให้ } F = 0 \\ (f_{avg} - c\sigma) - f & \text{สำหรับ MIN (f) ถ้า } F < 0 \text{ กำหนดให้ } F = 0 \end{cases} \quad (5.8)$$

โดยที่ c = จำนวนเท่าของส่วนส่วนเบี่ยงเบนมาตรฐานที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\sigma = \sqrt{\frac{\sum_{i=1}^{popsize} (f_i - f_{avg})^2}{popsize}}$$

วิธีการปรับสัดส่วนแบบตัดส่วนเบี่ยงเบนมาตรฐานเป็นการปรับค่าความเหมาะสมให้ลดลงเป็นระยะ c เท่าของส่วนเบี่ยงเบนมาตรฐานจากค่าความเหมาะสมเฉลี่ย ดังนั้นจึงเป็นการพิจารณาโครโมโซมที่ไม่ดี เฉพาะโครโมโซมที่ไม่แตกต่างจากโครโมโซมที่มีความเหมาะสมเฉลี่ยเกินกว่าระยะ c เท่าของส่วนเบี่ยงเบนมาตรฐาน ซึ่งโครโมโซมที่ไม่ดีที่เบี่ยงเบนไปจากค่าความเหมาะสมเฉลี่ยมากเกินไป หรือแตกต่างจากโครโมโซมส่วนใหญ่หลายๆ จะถูกตัดทิ้งโดยกำหนดให้มีค่าความเหมาะสมเป็น 0 ซึ่งจะทำการปรับค่าความเหมาะสมไม่มีค่าที่เป็นลบเกิดขึ้น สำหรับการกำหนดค่า c ที่ดีควรมีค่าอยู่ในช่วง $[1.0, 3.0]$ ขึ้นอยู่กับการประยุกต์ใช้ในแต่ละปัญหา เช่น การหาค่าสูงสุดของปัญหาซึ่งประกอบด้วยโครโมโซมที่มีค่าความเหมาะสมคือ 900, 615, 320 และ 595 ซึ่งกำหนดให้ $c = 1.2$ แล้วปรับส่วนเบี่ยงเบนมาตรฐานดังนี้

ลำดับ	ไม่ปรับสัดส่วน			ปรับสัดส่วน				
	ค่าความเหมาะสม (F=f)	ค่าความน่าจะเป็น (pselect _i)	จำนวนที่คาดหวัง (E _i)	ค่าความเหมาะสม (F=f - (f _{avg} - cσ))	ค่าความน่าจะเป็น (pselect _i)	ค่าความเหมาะสม (F=f - (f _{avg} - cσ))	ค่าความน่าจะเป็น (pselect _i)	จำนวนที่คาดหวัง (E _i)
1	900	0.3704	1.4815	576.8308	0.5072	576.8308	0.5058	2.0231
2	615	0.2531	1.0123	291.8308	0.2566	291.8308	0.2559	1.0235
3	20	0.1317	0.5267	-3.1692	-0.0028	0.0000	0.0000	0.0000
4	595	0.2449	0.9794	271.8308	0.2490	271.8308	0.2383	0.9534
รวม	2430					1140.4924		

$$f_{avg} = 607.5$$

$$\sigma = 236.9423$$

ตาราง 5-5 แสดงค่าต่างๆ โดยใช้วิธีปรับสัดส่วนแบบตัดส่วนเบี่ยงเบนมาตรฐาน

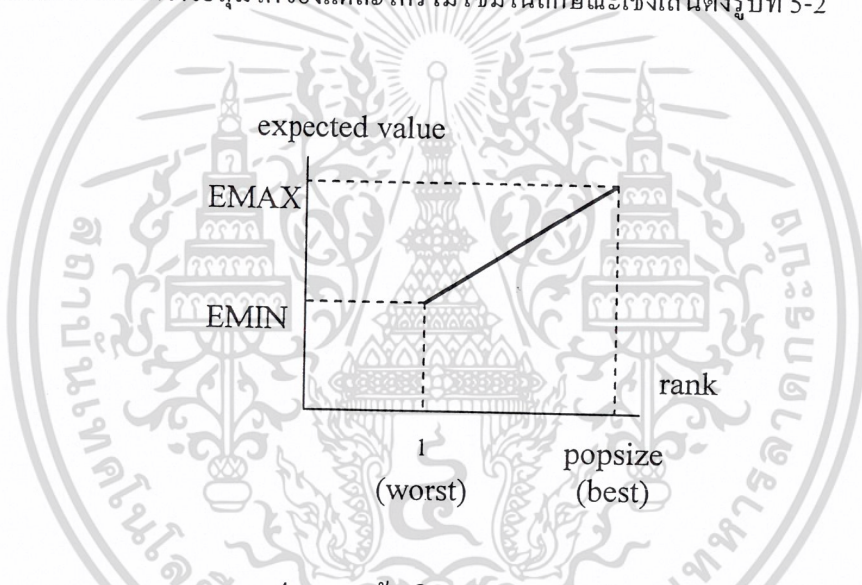
จะเห็นได้ว่าโครโมโซมลำดับที่ 3 มีค่าความเหมาะสมที่ต่ำกว่าค่าความเหมาะสมเฉลี่ยเกินกว่า 1.2 เท่าของส่วนเบี่ยงเบนมาตรฐาน ซึ่งทำให้ค่าความเหมาะสมที่ถูกปรับมีค่าเป็นลบจึงทำการตัดโครโมโซมนี้ทิ้งไป โดยการกำหนดเป็นจำนวนที่คาดหวังว่าจะสุ่มได้โครโมโซมอื่นแทนต่อไป

5.1.2 การคัดเลือกแบบอ้างอิงลำดับ (Ranking-based) เป็นอีกวิธีในการคัดเลือกโครโมโซมต้นแบบเพื่อทำการสร้างเมตติงพูล ที่จะช่วยลดจุดสิ้นสุดก่อนที่เป็นสาเหตุจากโครโมโซมที่ดีเกินไปโดยการควบคุมการจัดสรรที่คาดหวังว่าจะสุ่มได้ของแต่ละโครโมโซม ที่จะช่วยไม่ให้โครโมโซมใดถูกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดสรรจำนวนที่คาดหวังว่าจะสุ่มได้ในจำนวนที่มากขึ้นไป วิธีของการคัดเลือกแบบอ้างอิงลำดับนั้น จะไม่ได้พิจารณาค่าความเหมาะสมโดยตรง แต่จะกำหนดให้แต่ละโครโมโซมมีจำนวนที่คาดหวังว่าจะสุ่มได้จากการพิจารณาลำดับความสำคัญ (rank) ของค่าความเหมาะสมของโครโมโซมในแต่ละรุ่น

หลักการกำหนดจำนวนที่คาดหวังว่าจะสุ่มได้ควรจัดสรรในลักษณะจำนวนที่เพิ่มขึ้นตามประสิทธิภาพของค่าความเหมาะสมของแต่ละโครโมโซม และผลรวมของจำนวนที่คาดหวังว่าจะสุ่มได้ควรมีค่าเท่ากับจำนวนโครโมโซมที่จะเกิดขึ้นในรุ่นถัดไป ซึ่งโดยทั่วไปวิธีการคัดเลือกแบบอ้างอิงลำดับมี 2 วิธีคือ

5.1.2.1 แบบเชิงเส้น (Linear) เป็นวิธีการกำหนดจำนวนที่คาดหวังว่าจะสุ่มได้ของแต่ละโครโมโซมที่กำหนดสัดส่วนจากการอ้างอิงลำดับค่าความเหมาะสม วิธีการโดยเรียงลำดับความสำคัญของโครโมโซมที่ไม่ดีที่สุดให้เป็นลำดับที่ 1 (rank = 1) โครโมโซมที่ดีขึ้นกำหนดให้มีลำดับที่สูงขึ้น และกำหนดจำนวนที่คาดหวังว่าจะสุ่มได้โครโมโซมที่ดีที่สุดที่ต้องการ ซึ่งควรมีค่าอยู่ในช่วง [1.0,2.0] แล้วจึงจัดสรรจำนวนที่คาดหวังว่าจะสุ่มได้ของแต่ละโครโมโซมในลักษณะเชิงเส้นดังรูปที่ 5-2



รูป 5-2 การอ้างอิงลำดับแบบเชิงเส้น

ดังนั้นจำนวนที่คาดหวังว่าจะสุ่มได้ของแต่ละโครโมโซมคำนวณได้ดังสมการที่ 5-9

$$E_i = EMIN + (EMAX - EMIN) * \frac{rank_i - 1}{popsize - 1} \tag{5.9}$$

โดยที่ EMAX = จำนวนที่คาดหวังว่าจะสุ่มได้โครโมโซมที่ดีที่สุด (1 ≤ EMAX < 2)

EMIN = 2- EMAX

rank_i = ลำดับความสำคัญของโครโมโซมที่ i

เช่น การหาค่าสูงสุดของปัญหาซึ่งประกอบด้วยโครโมโซมที่มีค่าความเหมาะสมคือ 500,20,90 และ 15 ซึ่งกำหนดให้ $EMAX = 1.5$ แล้วเปรียบเทียบการอ้างอิงค่าความเหมาะสมกับการอ้างอิงลำดับความสำคัญค่าความเหมาะสมดังนี้

ลำดับโครโมโซม	อ้างอิงค่าความเหมาะสม			อ้างอิงลำดับค่าความเหมาะสม	
	ค่าความเหมาะสม ($F=f$)	ค่าความน่าจะเป็น ($pselect_i$)	จำนวนที่คาดหวัง (E_i)	ลำดับความสำคัญ (rank)	จำนวนที่คาดหวัง (E_i)
1	500	0.8000	3.2000	4	1.5000
2	20	0.0320	0.1280	2	0.8333
3	90	0.1440	0.5760	3	1.1667
4	15	0.0240	0.0960	1	0.5000
รวม	625				

ตาราง 5-6 แสดงการเปรียบเทียบการอ้างอิงค่าความเหมาะสมกับการอ้างอิงลำดับค่าความเหมาะสม

จะเห็นว่าโครโมโซมลำดับที่ 1 จะมีจำนวนที่คาดหวังว่าจะสุ่มได้ลดลงจาก 3.2 เป็น 1.5 หากอ้างอิงลำดับความสำคัญของค่าความเหมาะสมแทนซึ่งช่วยลดจำนวนต้นแบบในเมตดิงพลตลงได้

5.1.2.2 แบบไม่เชิงเส้น (Nonlinear) เป็นวิธีการกำหนดจำนวนที่คาดหวังว่าจะสุ่มได้ของแต่ละโครโมโซมจากการกำหนดค่าความน่าจะเป็นที่จะสุ่มได้โครโมโซมที่ดีที่สุดมีค่าอยู่ในช่วง $[0.0,1.0]$ โดยเรียงลำดับความสำคัญของโครโมโซมที่ดีที่สุดคือลำดับที่ 1 และโครโมโซมที่ไม่ดีมีลำดับสูงขึ้น ซึ่งจำนวนที่คาดหวังว่าจะสุ่มได้แต่ละโครโมโซมคำนวณได้ดังสมการที่ 5.10

$$E_i = \text{popsize} * (1 - q)^{\text{rank}_i - 1} \quad (5.10)$$

โดยที่ rank_i = ลำดับความสำคัญของโครโมโซมที่ i

q = ค่าความน่าจะเป็นที่จะสุ่มได้โครโมโซมที่ดีที่สุดในแต่ละครั้ง

ค่า q เป็นค่าที่ใช้ในการกำหนดความน่าจะเป็นที่จะสุ่มได้โครโมโซมที่ดีที่สุดในแต่ละครั้ง ซึ่งขึ้นอยู่กับการประยุกต์ใช้ในแต่ละปัญหาเช่น หากกำหนดให้ $q = 0.04$ ซึ่ง $\text{popsize} = 50$ และจัดลำดับความสำคัญของโครโมโซมได้เป็น $[5,11,2..]$ ดังนั้นค่า E_i ของแต่ละโครโมโซมคือ $E_1 = 50 \times 0.04 \times 0.96^{(1-1)} = 1.33$, $E_3 = 50 \times 0.04 \times 0.96^{(2-1)} = 1.92, \dots$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 วิธีการสุ่มต้นแบบ

ในการสร้าง เมตดิงฟูล โดยการสุ่มเลือกโครโมโซมแบบตามค่าความน่าจะเป็นที่จะสุ่มได้ของแต่ละโครโมโซม ซึ่งควรสุ่มให้ได้จำนวนโครโมโซมต้นแบบให้ถูกต้องตามจำนวนที่คาดหวังว่าจะสุ่มได้ของแต่ละโครโมโซม โดยทั่วไปมี 3 วิธีคือ

5.2.1 จำลองแบบการหมุนวงล้อ (Roulet Wheel : RW) เป็นวิธีการสุ่มของ เจเนติก อัลกอริทึม ในยุคแรกๆ คั้งที่ได้กล่าวมาแล้ว ในบทที่ 3

5.2.2 สุ่มทศนิยมแบบคืนกลับ (Stochastic Remainder Sampling with Replacement : SR) เนื่องจากจำนวนของค่าที่คาดหวังว่าจะสุ่มได้มักจะมีค่าที่ไม่เป็นจำนวนเต็ม ดังนั้นการเพิ่มความถูกต้องของการสุ่มที่ใช้ในการสร้างเมตดิงฟูล สามารถทำได้โดยการใช้การสุ่มทศนิยมแบบคืนกลับ ซึ่งมีขั้นตอนดังนี้

ขั้นที่ 1 : กำหนดให้แต่ละโครโมโซมถูกจัดสรรให้เป็นโครโมโซมต้นแบบเป็นจำนวนเท่ากับจำนวนที่คาดหวังว่าจะสุ่มได้ของแต่ละโครโมโซม

ขั้นที่ 2 : เมตดิงฟูลส่วนที่เหลือจะแทนด้วยโครโมโซมที่ได้จากการสุ่มจากค่าทศนิยมของจำนวนที่คาดหวังว่าจะสุ่มได้ของแต่ละโครโมโซมนั้น ด้วยวิธีการจำลองแบบการหมุนวงล้อ ซึ่งแต่ละช่วงของวงล้อจะถูกกำหนดตามค่าทศนิมนั้นๆ

เช่น สุ่มโครโมโซมต้นแบบของโครโมโซมที่มีค่าความเหมาะสมเป็น 900, 240, 575 และ 570 ดังนี้

ลำดับโครโมโซม	ค่าความเหมาะสม (F=f)	ค่าความน่าจะเป็น (pselect _i)	จำนวนที่คาดหวัง (E _i)	จำนวนต้นแบบจากค่าจำนวนเต็ม	จำนวนต้นแบบจากการสุ่มค่าทศนิยม	จำนวนต้นแบบในเมตดิงฟูล
1	900	0.3939	1.5755	1	1	2
2	240	0.1050	0.4201	0	0	0
3	575	0.2516	1.0066	1	0	1
4	570	0.2495	0.9978	0	1	1
รวม	2285					

ตาราง 5-7 แสดงการสุ่มเลือกโครโมโซมต้นแบบโดยวิธีสุ่มทศนิยมแบบคืนกลับ

จำนวนของโครโมโซมต้นแบบในเมตดิงฟูล ของโครโมโซมลำดับที่ 1 และ 4 จากจำนวนเต็มของค่าจำนวนที่คาดหวัง เป็น 1 โครโมโซม และสุ่มอีก 2 โครโมโซม โดยทำการสุ่มค่าทศนิยมของจำนวนที่คาดหวังของแต่ละโครโมโซมด้วยวิธีการหมุนวงล้ออีกดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับโครโมโซม	1	2	3	4
ค่าทศนิยม	0.5755	0.4201	0.0066	0.9978
ค่าความน่าจะเป็นที่จะสุ่มได้ในแต่ละครั้ง (p_{select_i})	0.2813	0.2120	0.0033	0.5034
ความถี่สะสมของค่าความน่าจะเป็น (q_i)	0.2813	0.4933	0.4966	1.0000
สร้างเลขสุ่มในการหมุนวงล้อแต่ละครั้ง (r)	0.8331	0.1446		
ลำดับโครโมโซมที่ถูกเลือก ($q_{i-1} \leq r \leq q_i$)	4	1		

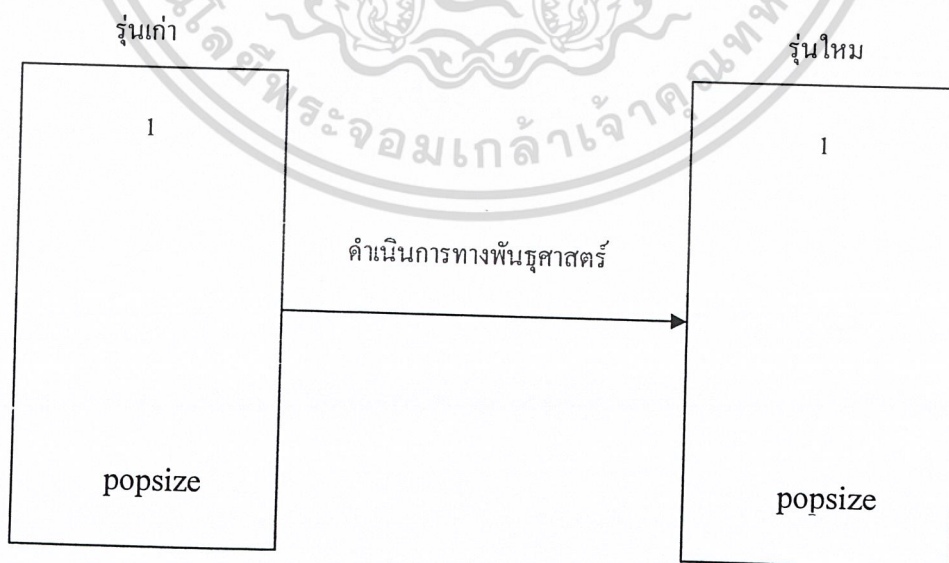
ตาราง 5-8 แสดงการสุ่มเลือกโครโมโซมต้นแบบโดยวิธีหมุนวงล้อด้วยค่าทศนิยมของค่าความคาดหวัง

โครโมโซมต้นแบบในเมตติงพูลที่เหลืออยู่จากการสุ่มค่าทศนิยมด้วยวิธีการหมุนวงล้อ คือ โครโมโซมลำดับที่ 4 และ 1 ทำให้จำนวนของโครโมโซมต้นแบบทั้งหมดเป็น 2, 0, 1 และ 1 โครโมโซมตามลำดับ

5.3 การรีโพรดักชัน (Reproduction)

การรีโพรดักชันเป็นกระบวนการที่ทำให้เกิดประชากรรุ่นใหม่จากการถ่ายทอดโครโมโซมรุ่นเก่าที่ผ่านขั้นตอนต่างๆ มา การรีโพรดักชันมีหลายวิธีได้แก่

5.3.1 รีโพรดักชันแบบทั่วไป เป็นการรีโพรดักชันของ เจเนติกอัลกอริทึม ในยุคแรกๆ โดยการถ่ายทอดโครโมโซมรุ่นเก่าที่ผ่านกระบวนการต่างๆ ทั้งหมดกลายเป็นประชากรรุ่นใหม่ตามจำนวนประชากรในแต่ละรุ่นที่กำหนดคงรูป ซึ่งชุดโครโมโซมรุ่นใหม่อาจมีค่าดอปที่ไม่ดีเท่ากับชุดโครโมโซมรุ่นเก่า เนื่องจากการดำเนินการของ เจเนติกอัลกอริทึม ไม่ได้มีการบันทึกโครโมโซมที่ดีเอาไว้ ดังนั้นการทำการรีโพรดักชันแบบนี้อาจทำให้สูญเสียโครโมโซมที่ดีที่ใช้เป็นต้นแบบในการสร้างโครโมโซมที่ดีขึ้นไป

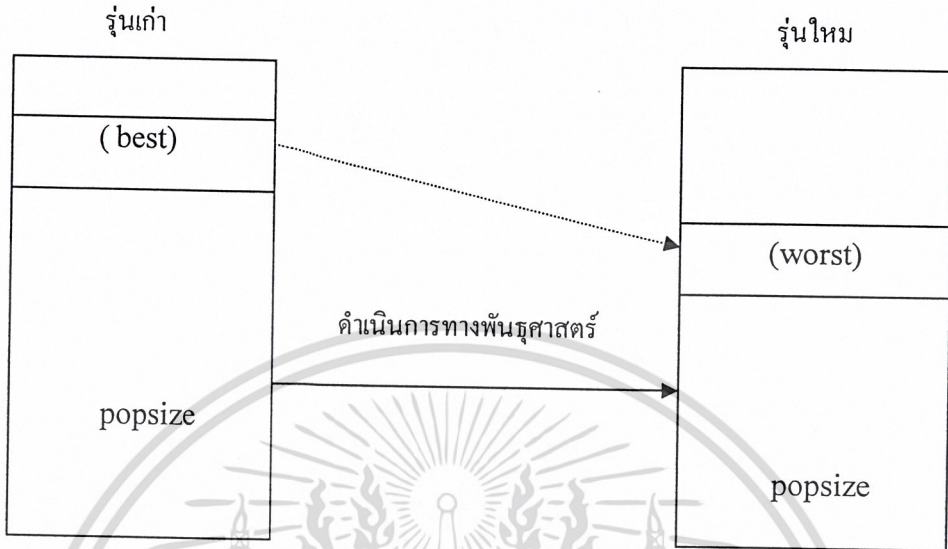


รูป 5-3 การรีโพรดักชันแบบทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

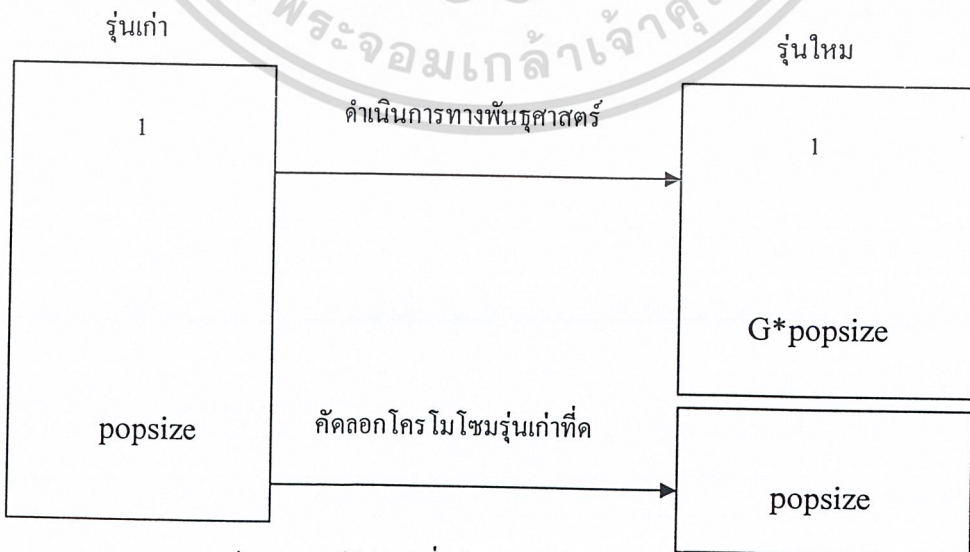
5.3.2 รีโปรดักชันโดยการรักษาโครโมโซมที่ดีที่สุดเอาไว้ (Elitism) เป็นการปรับปรุงการรีโปรดักชันแบบทั่วไปด้วยการรักษาโครโมโซมที่ดีที่สุดจากรุ่นเก่าเอาไว้ โดยทำการรีโปรดักชันแบบทั่วไปก่อน แล้วค่อยทำการคัดลอกโครโมโซมที่ดีที่สุดจากรุ่นเก่ามาใส่แทนโครโมโซมที่แย่ที่สุดในรุ่นใหม่ ดังรูปที่

5-4



รูป 5-4 การรีโปรดักชัน โดยการรักษาโครโมโซมที่ดีที่สุดเอาไว้

5.3.3 รีโปรดักชันโดยรักษาสถานะคงที่แบบซ้ำ (Steady State with Duplicate Reproduction : SS) โดยการกำหนดอัตราการสร้างโครโมโซมรุ่นใหม่ (Generation Gap : G) โดยที่ $1 < G \leq 0$ สำหรับการกำหนดค่า G ที่เหมาะสมนั้นจะขึ้นอยู่กับแต่ละปัญหา ซึ่งจำนวนโครโมโซมที่จะสร้างขึ้นมาใหม่นั้นจะเท่ากับ $popsize * G$ (ถ้า $G = 1$ คือ การรีโปรดักชันแบบทั่วไป) และโครโมโซมที่เหลือเกิดจากการคัดลอกกลุ่มโครโมโซมที่ดีจากรุ่นเก่าตามลำดับของความสำคัญของโครโมโซมนั้น ซึ่งจะทำให้โครโมโซมที่ดีจากรุ่นเก่าสามารถมีโอกาสรอดจากรุ่นหนึ่งไปยังอีกรุ่นหนึ่งเพื่อนำมาเป็นโครโมโซมต้นแบบได้มากขึ้น

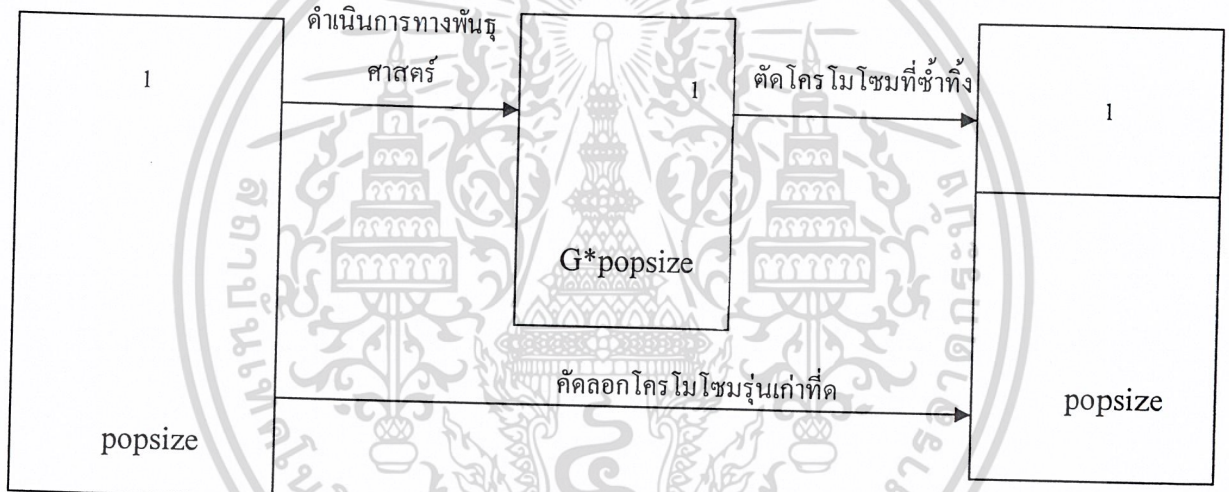


รูป 5-5 การรีโปรดักชัน โดยการรักษาสถานะแบบซ้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.4 รีโพรดักชันโดยการรักษาสถานะคงที่แบบไม่ซ้ำ (Steady State without Duplicate Reproduction : SS W/O) เนื่องจากในการดำเนินการของ เจเนติกอัลกอริทึม นั้น โครโมโซมรุ่นใหม่ที่เกิดขึ้นในขั้นตอนต่างๆของแต่ละโครโมโซมนั้น อาจมีโอกาสเกิดโครโมโซมที่มีความเหมือนกัน หรือ ซ้ำกันกับโครโมโซมรุ่นเก่าได้ นั่นคือจำนวนที่คาดหวังว่าจะสุ่มได้ของโครโมโซมนั้นจะเพิ่มขึ้นเป็นทวีคูณตามจำนวนที่ซ้ำกัน ซึ่งทำให้ เจเนติกอัลกอริทึม ต้องใช้เวลาดำเนินการกับโครโมโซมที่ซ้ำกันเหล่านั้นตลอดการรีโพรดักชัน โดยการรักษาสถานะคงที่แบบไม่ซ้ำจะช่วยให้การค้นหาคำตอบของ เจเนติกอัลกอริทึม เกิดรูปแบบโครโมโซมต่างๆ มากขึ้น

วิธีการนี้มีความคล้ายคลึงกับการรีโพรดักชันโดยการรักษาสถานะคงที่แบบซ้ำ คือ มีการรักษา กลุ่มของโครโมโซมที่ดีในรุ่นเก่าเอาไว้ และมีการตัดโครโมโซมรุ่นใหม่ที่ซ้ำกับโครโมโซมรุ่นเก่าที่ทำการคัดลอกมาออกไป ดังนั้น โครโมโซมในแต่ละรุ่นที่ทำการสร้างขึ้นจะไม่เกิดการซ้ำกับโครโมโซมในรุ่นก่อนเลย



รูป 5-6 การรีโพรดักชันโดยการรักษาสถานะแบบไม่ซ้ำ

บทสรุป

การพัฒนาเจเนติกอัลกอริทึมแบบต่างๆ โดยการเพิ่มเทคนิคต่างๆ ตามที่ได้กล่าวมาแล้วเข้าไปสามารถช่วยปรับปรุงประสิทธิภาพการทำงานของเจเนติกอัลกอริทึมได้เป็นอย่างดี ซึ่งจากการค้นคว้าและการทดลองใช้ เจเนติกอัลกอริทึมกับปัญหาต่างๆพบว่า โดยเฉลี่ยแล้วการรีโพรดักชันจะทำให้การค้นหาคำตอบของเจเนติกอัลกอริทึมดีขึ้น เพราะจำนวนโครโมโซมที่ถูกสร้างขึ้นมาใหม่ และจำนวนโครโมโซมที่ดีในรุ่นเก่าที่รักษาไว้เป็นสัดส่วนที่เหมาะสมกับปัญหาเหล่านั้น สำหรับการปรับสัดส่วนค่าความเหมาะสมไม่ว่าจะทำด้วยวิธีใดก็ตามโดยเฉลี่ยแล้วจะช่วยให้การทำงานของเจเนติกอัลกอริทึมดีขึ้นกว่าการที่ไม่ทำการปรับสัดส่วนใดๆ เลย

บทที่ 6

การออกแบบ และพัฒนาแอปพลิเคชัน

ในบทนี้จะนำเสนอเกี่ยวกับการออกแบบและพัฒนาแอปพลิเคชัน ในส่วนของโปรแกรมที่พัฒนาโดยใช้ Microsoft Visual Studio.NET C# ซึ่งเป็นโปรแกรมส่วนที่ใช้สร้างกราฟฟิคยูสเซอร์อินเทอร์เฟซ (Graphic User Interface : GUI) และทำการคำนวณเกี่ยวกับพอร์ตทั้งหมดซึ่งมีรายละเอียดดังต่อไปนี้

6.1 อินพุตและเอาต์พุตของแอปพลิเคชัน

อินพุตของแอปพลิเคชันนี้มาจาก 2 ส่วนด้วยกัน คือ

(1) นักลงทุน ซึ่งจะให้อินพุตต่างๆแก่แอปพลิเคชันผ่านการอัปโหลด Excel file โดยอินพุตในส่วนนี้ประกอบด้วย

- วันที่ต้องการคำนวณ (Position Date)
- ราคาของพอร์ตที่ต้องการ
- กลุ่มของหุ้นที่นักลงทุนสนใจจะลงทุน
- จำนวนหุ้นที่ต้องการเป็นคำตอบจากแอปพลิเคชัน
- Duration ของพอร์ตที่นักลงทุนต้องการ
- จำนวนครั้งที่ต้องการให้คำนวณ

(2) ฐานข้อมูล ซึ่งจะให้อินพุตกับแอปพลิเคชันดังนี้

- อัตราผลตอบแทนที่ได้ในแต่ละงวด(Payment)
- งวดที่ต้องจ่ายอัตราผลตอบแทน (CouponDate) โดยจะเก็บเป็นจำนวนวัน ดังแสดงใน

รูป 6-1 และ 6-2

PORTFOLIO MANAGEMENT

▶ Portfolio Management Model :

▶ Position Date (dd/mm/yyyy) วันที่จะทำการคำนวณ
Minimum available date is 30/06/2004.

ราคาของ Port ที่ต้องการ Portfolio Value : Baht Number of Bond จำนวนของหุ้นที่ต้องการเลือกเข้า Port

Portfolio Duration : year Number of Calculation (1 -10)
Duration ของ Port ที่ต้องการ ต้องการให้คำนวณจำนวนกี่ครั้ง

▶ Please enter your trading position data:
 Only registered corporate bonds are allowed (you may [Download](#) a sample excel file for formatting).

รูป 6-1 แสดงหน้าอินพุตของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORTFOLIO MANAGEMENT

▶ Portfolio Management Model : Immunization ▼

▶ Position Date : 30/06/2004 (dd/mm/yyyy)
Minimum available date is 30/06/2004.

Portfolio Value : 100000 Baht Number of Bond : 4

Portfolio Duration : 1.1 year Number of Calculation : 2 (1 -10)

▶ Please enter your trading position data:

Only registered corporate bonds are allowed (you may Download a sample excel file for formatting).

upload ข้อมูล

ทำการเลือก excel file ซึ่งมี set ของหุ้นกู้
ที่นักลงทุนมีความสนใจและยอมรับที่จะลงทุน

<input type="checkbox"/>	Order #	Symbol	Error Status	Edit
<input type="checkbox"/>	1	AIS063A		<input type="button" value="Edit"/>
<input type="checkbox"/>	2	LH06DA		<input type="button" value="Edit"/>
<input type="checkbox"/>	3	BP064A		<input type="button" value="Edit"/>
<input type="checkbox"/>	4	CPF063A		<input type="button" value="Edit"/>

เชคของหุ้นกู้ที่นักลง
ทุนมีความสนใจและ
ต้องการให้ทำการ
เลือก

submit ข้อมูลไปทำการคำนวณ

รูป 6-2 แสดงหน้าอินพุทของแอปพลิเคชันหลังจากมีการอัปโหลดข้อมูล

เอาต์พุทของแอปพลิเคชันนี้ประกอบด้วยหน้าเอาต์พุตต่างๆดังต่อไปนี้

(1) หน้าเอาต์พุตที่แสดง ราคา Duration และ Convexity ของหุ้นกู้ทุกตัวที่อยู่ในเซตที่ผู้ใช้

ต้องการ



Calculation made on 2 February 2005.

Symbol	Price	Duration	Convexity
AIS063A	847.623496	1.057623	2467.942346
LH06DA	849.172094	1.479457	5931.641739
BP064A	848.481955	1.325284	4446.429408
CPF063A	868.23005	1.060537	3323.952812

รูป 6-3 แสดงหน้าเอาต์พุตหน้าที่ 1

(2) พอร์ตที่ประกอบด้วยกลุ่มของหุ้นกู้ที่ให้ Duration รวมใกล้เคียงกับ Duration ที่นักลงทุน
ต้องการที่สุด โดยจะแสดงเอาต์พุตของการคำนวณเป็นจำนวนครั้งตามที่ผู้ใช้ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

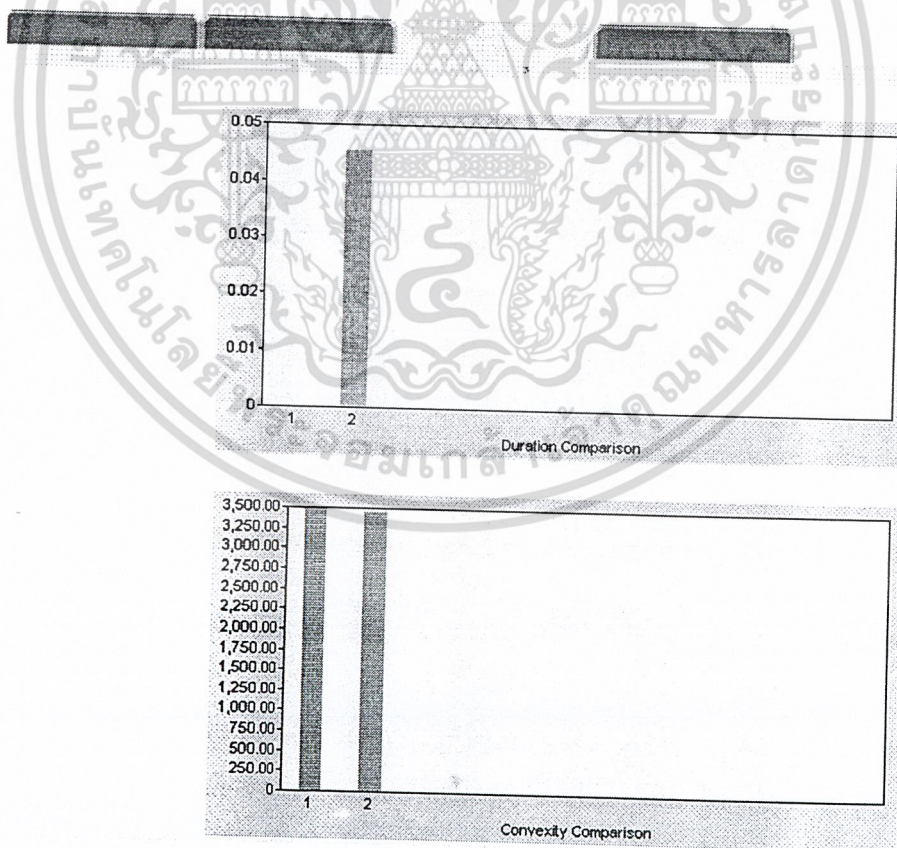
Calculation made on 2 February 2005.
Required Duration is 1.1 Year.

Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	9.110232	5900	0.0624	145.608598
LH06DA	6.830182	5800	0.095809	344.035221
BP064A	6.835738	5800	0.076866	257.892906
CPF063A	95.020899	82500	0.874943	2742.26107
Total Value	117.797051	100000	1.100018	3489.797795

Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	78.131816	50600	0.535157	1248.778827
LH06DA	24.023399	20400	0.301809	1210.054915
BP064A	2.357151	2000	0.026506	88.928588
CPF063A	31.097749	27000	0.286345	897.467259
Total Value	135.610115	100000	1.149817	3445.229593

รูป 6-4 แสดงหน้าเอาท์พุทหน้าที่ 2

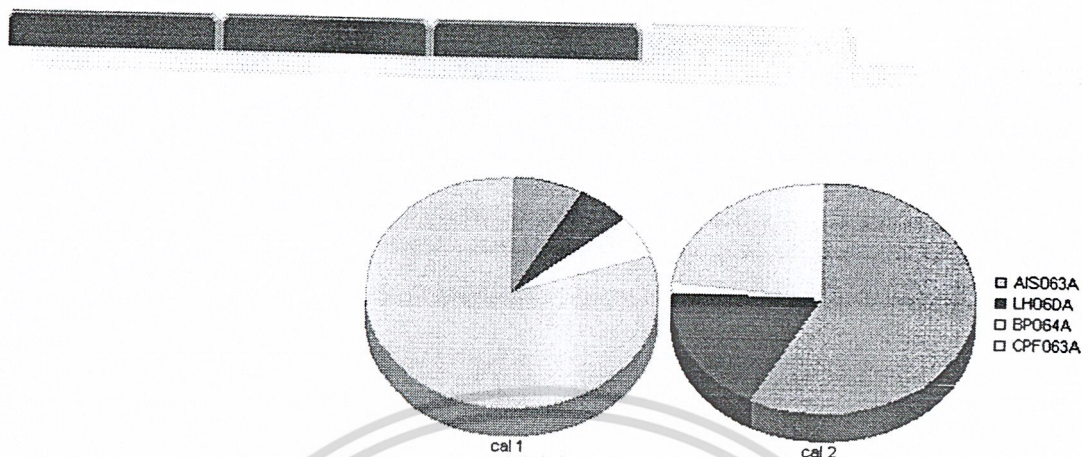
(3) หน้าเอาท์พุทที่แสดงค่าความเบี่ยงเบนของการคำนวณแต่ละครั้งว่า แต่ละครั้งการคำนวณมีค่า Duration เบี่ยงเบนไปจาก Duration ที่ต้องการเท่าใด



รูป 6-5 แสดงหน้าเอาท์พุทหน้าที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(4) หน้าเอาที่พื้ที่แสดงกราฟิกของการคำนวณว่าในแต่ละครั้งต้องซื้อหุ้นกี่ตัวใดบ้าง ตัวละกี่เปอร์เซ็นต์ เมื่อเทียบกับทั้งพอร์ต



รูป 6-6 แสดงหน้าเอาที่พื้หน้าที่ 4

โดยแอปพลิเคชันนี้จะมีคลาสต่างๆซึ่งเป็นส่วนประกอบในการทำงานอยู่ 2 แพ็กเกจหลักด้วยกัน คือ

1. แพ็กเกจ PortfolioManagement คือ แพ็กเกจที่มีลอจิกการทำงานของแอปพลิเคชันทั้งหมด โดยทำงานของแพ็กเกจนี้จะดูแลตั้งแต่การรับอินพุต การตรวจสอบการถูกต้องของอินพุต คิดต่อถึงข้อมูลที่จำเป็นจากฐานข้อมูล แล้วเรียกใช้กระบวนการเจเนติกอัลกอริทึมจนกระทั่งได้คำตอบและนำคำตอบไปแสดงผล
2. แพ็กเกจ GeneticAlgorithm คือ แพ็กเกจที่มีลอจิกการทำงานเฉพาะในส่วนของการเจเนติกอัลกอริทึม เช่น มีกระบวนการในการครอสโอเวอร์และการมิวเตชัน มีกระบวนการในการทำรีโพรดักชันการสุ่มเลือกต้นแบบ เป็นต้น

6.2 โครงสร้างของแพ็กเกจ Portfolio Management

แพ็กเกจ Portfolio Management ประกอบด้วยแพ็กเกจย่อยๆ ได้แก่

6.2.1 แพ็กเกจ PortfolioManagement.Bizlogic

ประกอบด้วยคลาสย่อย คลาส คือ

1. คลาส InputValidation ทำหน้าที่ในการตรวจสอบความถูกต้องของอินพุต
2. คลาส PortfolioManagement ทำหน้าที่ควบคุมขั้นตอนการทำงานทั้งหมด หลังจากที่ได้รับมาจากการตรวจสอบว่าถูกต้องทั้งหมดแล้ว ซึ่งมีโครงสร้างดังรูป 6-7
3. คลาส BondDataListProcess ทำหน้าที่เป็นนำข้อมูลที่ดึงมาจากฐานข้อมูล (ดึงมาโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ฟังก์ชันของคลาสในแพ็คเกจ PortfolioManagement.Data) มาจัดให้อยู่ ในรูปแบบที่ต้องการ ซึ่งก็คือ การนำข้อมูลดิบที่ดึงมาได้จากฐานข้อมูลมาจัดให้อยู่ในรูป อ็อบเจก BondData ซึ่งมี คุณสมบัติต่างๆ ในการแทนหุ้นกู้ 1 ตัวอย่างครบถ้วน เช่น เงินปันผลในแต่ละงวด ค่า Duration ค่า convexity และมีโครงสร้างตามรูปที่ 6-8

```

namespace PortfolioManagement.Bizlogic

public class PortfolioManagement{

public Bizlogic(){}

public portManagementProcess ()
{
//เรียกใช้งานคลาสใน PortfolioManagement.Data
QueryDatabase();
//จัดรูปแบบของข้อมูลดิบจากฐานข้อมูลให้อยู่ในรูปอ็อบเจก
ManipulateRawData();
//คำนวณค่าคุณสมบัติต่างๆของหุ้นกู้
Calculation();
//เรียกใช้กระบวนการเจเนติกอัลกอริทึม
GeneticBiz.init();
//จัดรูปแบบส่วนแสดงผลโดยใช้ฟังก์ชันในคลาส Presentation
Presentation();
}
}

```

รูป 6-7 แสดงโครงสร้างของคลาส Bizlogic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public class BondData
{
    private string _SYMBOL;
    private double _UNIT;
    private double _DURATION;
    private double _PRICE;
    private double _CONVEXITY;
    private double _YIELD;
    private double[] _PAYMENT;
    private double[] _COUPONDATE;

    public string SYMBOL
    {
        get { return _SYMBOL;}
        set { _SYMBOL = value;}
    }

    public double UNIT
    {
        get { return _UNIT;}
        set { _UNIT = value;}
    }

    public double DURATION
    {
        get { return _DURATION;}
        set { _DURATION = value;}
    }
    .....
}

```

รูป 6-8 แสดงโครงสร้างของคลาส BondData

6.2.2 แพ็กเกจ PortfolioManagement.Calculation

ทำหน้าที่ในการคำนวณค่าซึ่งเป็นคุณสมบัติของหุ้นกู้ทั้งหมด เช่น ราคาของหุ้นกู้ duration และ convexity ของหุ้นกู้ก่อนที่จะใช้เงินเดกอัลกอริทึมในการหาค่าตอบ เนื่องจากในกระบวนการเงินเดกอัลกอริทึม จะต้องใช้ค่าเหล่านี้ในการคำนวณหาค่าความเหมาะสมในฟังก์ชันความเหมาะสมและฟังก์ชันจุดประสงค์ โดยการคำนวณนี้จะมีการคำนึงถึงความเสี่ยงของหุ้นแต่ละตัวจากอัตราความเสี่ยงของหุ้นกู้ด้วย

6.2.3 แพ็กเกจ PortfolioManagement.Data

ทำหน้าที่ในการดึงข้อมูลที่เป็นต้องใช้ในการคำนวณจากฐานข้อมูล เช่น วันที่จะทำการจ่ายเงินปันผลซึ่งจะมีการกำหนดไว้แล้ว ค่าเงินปันผลในแต่ละงวด เนื่องจากมีความจำเป็นต้องใช้ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหล่านี้ในการคำนวณค่า Convexity และ Duration ซึ่งเป็นค่าที่จำเป็นต้องใช้งานในกระบวนการเจเนติก อัลกอริทึมต่อไป

6.2.4 แพ็กเกจ PortfolioManagement.Presentation

ทำหน้าที่ในการจัดรูปแบบของข้อมูลให้อยู่ในรูปแบบที่เหมาะสมเพื่อนำไปแสดงผลที่หน้าจอ เอาท์พุท ทำการสร้างตารางที่ใช้ในการแสดงผล และสร้างกราฟในหน้าจอเอาท์พุท

6.3 โครงสร้างของแพ็กเกจ GeneticAlgorithm

แพ็กเกจ GeneticAlgorithm ประกอบด้วยแพ็กเกจย่อย ๆ ได้แก่

6.3.1 แพ็กเกจ GeneticAlgorithm.GeneticStruct

แพ็กเกจ GeneticAlgorithm.GeneticStruct ประกอบด้วยคลาสซึ่งบ่งบอกโครงสร้างของโครโมโซม ได้แก่

- คลาส Chromosome ซึ่งบ่งบอกโครงสร้างของโครโมโซม

6.3.1.1 โครงสร้างของคลาส Chromosome

```
public class Chromosome
{
    private double[] _UNITS;
    private double _dFITNESS;
    private double _cFITNESS;

    public Chromosome(int numOfGenes)
    {
        _UNITS = new double[numOfGenes];
    }

    public double[] UNITS
    {
        get {return _UNITS;}
        set {_UNITS = value;}
    }

    public double dFITNESS
    {
        get {return _dFITNESS;}
        set {_dFITNESS = value;}
    }

    public double cFITNESS
    {
        get {return _cFITNESS;}
        set {_cFITNESS = value;}
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public Chromosome clone()
{
    Chromosome newch = new
    Chromosome(this.UNITS.Length);

    for(int i = 0; i < this.UNITS.Length ; i ++ )

        newch.UNITS[i] = this.UNITS[i];

    newch.dFITNESS = this.dFITNESS;
    newch.cFITNESS = this.cFITNESS;

    return newch;
}
}

```

รูป 6-9 แสดงรายละเอียดของคลาส Chromosome

ในส่วนคุณสมบัติของ คลาส chromosome นั้น จะประกอบด้วย

- จำนวนหน่วยของหุ้นกู้แต่ละตัวในพอร์ตซึ่งนับเป็นยีนแต่ละตัวนั่นเอง
- ค่า Duration ของโครโมโซม
- ค่า Convexity ของโครโมโซม

6.3.2 แพ็กเกจ GeneticAlgorithm.GeneticBiz

แพ็กเกจ GeneticAlgorithm.GeneticBiz ประกอบด้วยคลาส

1. คลาส GeneticBiz ซึ่งเป็นคลาสที่มี method ในการควบคุมการทำงานของกระบวนการเจเนติกอัลกอริทึมทั้งหมดตั้งแต่เริ่มต้นสร้างคอมบิเนชัน (Combination) ของหุ้นกู้ที่เป็นไปได้ทั้งหมด สร้างประชากรโครโมโซมรุ่นแรก และเรียกใช้กระบวนการเจเนติกอัลกอริทึม เพื่อหาค่าจำนวนหน่วยของหุ้นกู้แต่ละตัวที่เป็นชุดของคำตอบที่ดีที่สุด (โครโมโซมที่มีค่าความเหมาะสมที่สุด) สำหรับคอมบิเนชันนั้น เมื่อทำกระบวนการเจเนติกอัลกอริทึมกับทุกๆคอมบิเนชันแล้วจึงทำการเลือกคอมบิเนชันที่ดีที่สุดเพื่อเป็นคำตอบ มีโครงสร้างดังรูปที่ 6-10

2. คลาส GenBondCombination เป็นคลาสที่จะทำการหาคอมบิเนชันของหุ้นกู้ทั้งหมดที่เป็นไปได้จากเซตของหุ้นกู้ที่นักลงทุนสนใจและทำการอัปโพลด์ขึ้นมา โดย คอมบิเนชันของหุ้นกู้นี้จะมีลักษณะเป็นอาร์เรย์ของอ็อบเจกต์ BondData ซึ่งมีคุณสมบัติของหุ้นกู้อย่างครบถ้วนเช่น

นักลงทุนทำการอัปโพลด์หุ้นกู้ที่นักลงทุนมีความสนใจจะลงทุนขึ้นมา 3 ตัว ดังนี้

LB04NA LB22NA AIS063A

นักลงทุนต้องการให้แอปพลิเคชันเลือกหุ้นกู้เข้าพอร์ตจำนวน 2 ตัว จะได้คอมบิเนชันของหุ้นกู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เป็นไปได้ทั้งหมดจำนวน 3 คอมบิเนชัน ดังนี้

1. LB04NA LB22NA
2. LB04NA AIS063A
3. LB22NA AIS063A

```

namespace GeneticAlgorithm.GeneticBiz

public class GeneticBiz{

    private InputParams inputObj;
    private BondDataList bondInput;
    private int population = 10;
    private int generation = 100;
    private double crossoverRatio = 0.7;
    private double mutationRatio = 0.01 ;
    private ArrayList inipool = new ArrayList();
    private ArrayList offspringPool = new ArrayList();
    private ArrayList bestChrOfeachGen = new ArrayList();
    private BondDataList bondComb;

    public void init(ArrayList bondinput,int len)
    {
        //สร้างคอมบิเนชันที่เป็นไปได้ทั้งหมด
        Genbondcombination();
        //สร้างโครโมโซมรุ่นแรก
        //หาคำตอบที่ดีที่สุดของแต่ละคอมบิเนชัน
        GeneticAlgorithmProcess(initialPopululaion)
        //เลือกคอมบิเนชันที่ดีที่สุดเป็นคำตอบ
    }
    public void GeneticAlgorithmProcess(ArrayList inipop)
    {
        GeneticEquation equ = new GeneticEquation();
        for(int i = 0 ; i < Generations;i++)
        {
            find fitness for each chromosome in
            matingpool using equ.fitnessFunction;

            for(int j = 0 ;j < population/2 ; j++)
            {
                choose parent chromosome by RoulettWheel;

                use genetic Operator to improve
                chromosomes fitness;
            }
            get optimal output by equ.objectivefunction;
            return optimal Output;
        }
    }
}

```

รูป 6-10 แสดงโครงสร้างของคลาส GeneticBiz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.3 แพ็กเกจ GeneticAlgorithm.GeneticEquation

แพ็กเกจ GeneticAlgorithm.GeneticEquation ประกอบด้วยคลาสซึ่งมี method เกี่ยวกับสมการการคำนวณต่างๆที่เกี่ยวกับเจเนติกอัลกอริทึมซึ่งได้แก่

- ฟังก์ชันความเหมาะสม (Fitness Function)
- ฟังก์ชันจุดประสงค์ (Objective Function)

โดย ฟังก์ชันความเหมาะสม ในโครงงานนี้คือ $f(x) = 1/\text{UserDuration} - \text{PortDuration}$ และฟังก์ชันจุดประสงค์ คือ $F(x) = \max(1/\text{UserDuration} - \text{PortDuration})$ โดยคลาส GeneticEquation นี้มีโครงสร้างดังรูป 6-11

```
namespace GeneticAlgorithm.GeneticEquation
{
    public class GeneticEqu
    {
        public GeneticEqu(){}
        public void pmFitnessFunction(BondDataList
            bonds,Chromosome chr,double ExpDuration)
        {
            double calculateDuration = 0;
            ArrayList tempBond = bonds.getBondDataList();
            for(int i = 0; i < tempBond.Count; i ++ )
            {
                //--calculate Duration--//
                calculateDuration +=
                    chr.UNITS[i]*calBond.TTM;
            }
            double diff = Math.Abs(ExpDuration -
                calculateDuration);
            if(diff != 0)
                chr.dFITNESS = Math.Round(1/diff,6);
            else
            {
                chr.dFITNESS = 1/0.000001;
            }
        }
        public Chromosome pmObjectiveFunction(ArrayList Gen)
        {
            //find the best chromosome in generation
            return (Chromosome)Gen[0];
        }
    }
}
```

รูป 6-11 แสดงโครงสร้างของคลาส GeneticEquation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.4 แพ็กเกจ GeneticAlgorithm.GeneticOperator

แพ็กเกจ GeneticAlgorithm.GeneticOperator ประกอบด้วยคลาสซึ่งมี method เกี่ยวกับตัวดำเนินการทางพันธุศาสตร์ซึ่งใช้ในการปรับปรุงค่าความเหมาะสมของโครโมโซมแต่ละตัวให้ดียิ่งขึ้น ซึ่งตัวดำเนินการเหล่านั้นได้แก่

- คrossover
- mutation

โดยคลาส GeneticOperator มีโครงสร้างดังรูป 6-12

```

namespace GeneticAlgorithm.GeneticOperator
public class GeneticOperator{
    private void rescaleUnit()
    {
        //ทำการ reweight หน่วยในโครโมโซมให้ได้เท่ากับ 1 เหมือนเดิม (100%) รายละเอียดในบทที่ 4
    }
    public Chromosome[] crossover(Chromosome ch1,Chromosome ch2)
    {
        randomly choose starting point of crossover;
        crossover using partially crossover ;
        reweight unit;
    }
    public Chromosome mutation(Chromosome ch)
    {
        randomly choose point of mutation;
        mutation using random mutation;
        reweight unit;
    }
}

```

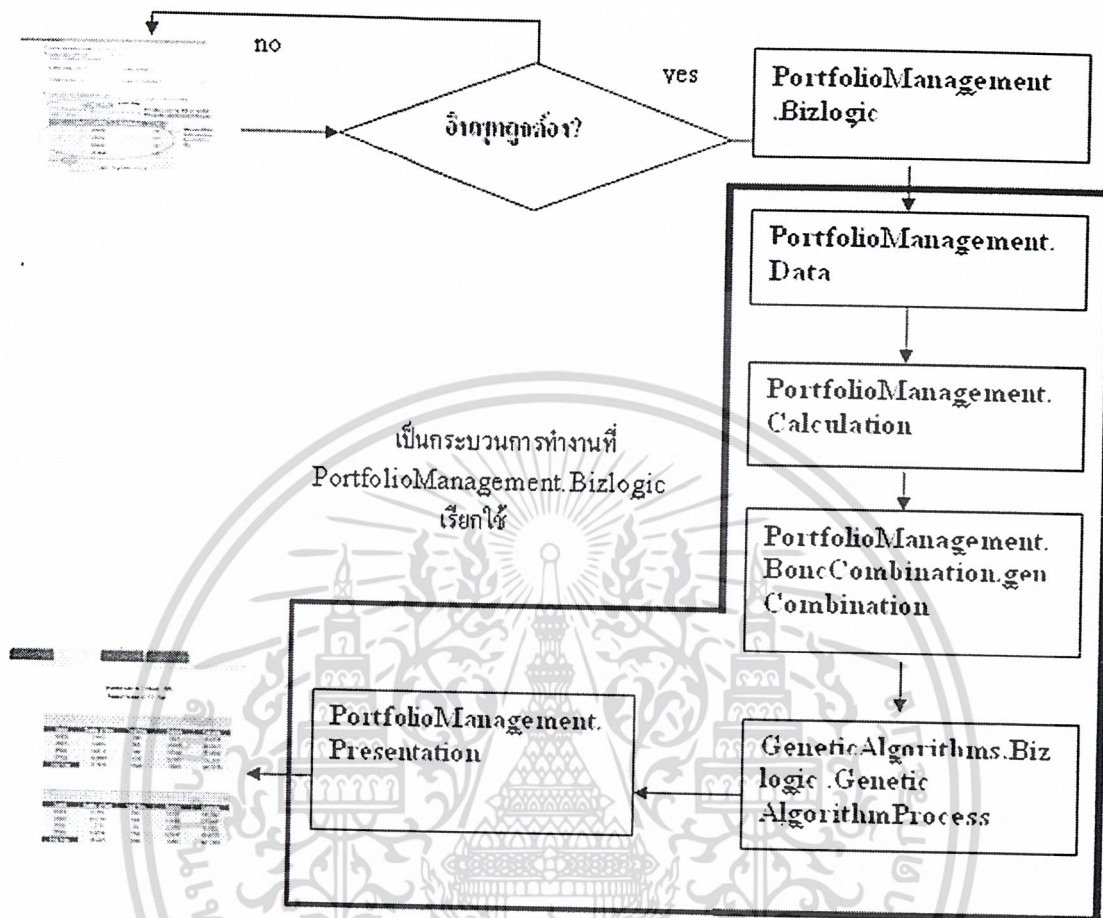
รูป 6-12 แสดงโครงสร้างของคลาส GeneticOperator

6.3.5 แพ็กเกจ GeneticAlgorithm.RoulettWheel

แพ็กเกจ GeneticAlgorithm.RoulettWheel ประกอบด้วยคลาสซึ่งมี method ในการคำนวณหาอัตราความเหมาะสม (fitness ratio) ของแต่ละโครโมโซมและนำค่าอัตราความเหมาะสมเหล่านั้นมาสร้าง RouletteWheel นอกจากนี้ยังมีฟังก์ชันที่ใช้ในการสุ่มเลือกต้นแบบในแต่ละครั้งของการรีโพรดักชันด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการทำงานของแอปพลิเคชันนี้จะมี Flow การทำงานดังรูปที่ 6-13



รูป 6-13 แสดงขั้นตอนการทำงานของแอปพลิเคชันระบบบริหารพอร์ตลงทุน

จากรูปที่ 6-13 สามารถอธิบายขั้นตอนการทำงานของแอปพลิเคชันได้ดังนี้

1. เริ่มต้นแอปพลิเคชันผู้ใช้จะทำการใส่ค่าอินพุตเข้ามาในระบบ
2. แอปพลิเคชันจะทำการตรวจสอบว่าค่าอินพุตทั้งหมดที่ได้รับมาจากผู้ใช้นั้นถูกต้องหรือไม่ โดยใช้ฟังก์ชันในคลาส InputValidation ในแพ็คเกจ PortfolioManagement.Bizlogic เพื่อทำการตรวจสอบถ้าอินพุตนั้นผิดพลาดก็จะให้ผู้ใช้ใส่ค่าอินพุตใหม่
3. จากนั้นจะทำการดึงข้อมูลที่จำเป็นต้องใช้ในการคำนวณจากฐานข้อมูล โดยใช้ฟังก์ชันการทำงานในแพ็คเกจ PortfolioManagement.Data และนำมาจัดรูปแบบให้อยู่ในรูปแบบ BondData
4. ทำการคำนวณค่าคุณสมบัติต่างๆของหุ้นกู้แต่ละตัว
5. จากนั้นทำการสร้างคอมบิเนชันที่เป็นไปได้ทั้งหมดจากเซตของหุ้นกูนักลงทุนสนใจจะลงทุน
6. ใช้เจเนติกอัลกอริทึมในการหาคำตอบที่ดีที่สุดของแต่ละคอมบิเนชัน
7. เลือกคอมบิเนชันที่ดีที่สุดมาเป็นคำตอบและนำไปแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทำงานแอปพลิเคชัน

การทำงานของแอปพลิเคชันนี้เป็นไปตามบทที่ได้กล่าวมาแล้ว
แอปพลิเคชันนี้จะปฏิบัติตามตัวอย่างด้านล่างนี้

โดยผลของการทำงานของ

ตัวอย่างที่ 1

PORTFOLIO MANAGEMENT

▶ Portfolio Management Model : Immunization

▶ Position Date: 30/06/2004 (dd/mm/yyyy)
Minimum available date is 30/06/2004.

Portfolio Value : 1000000 Baht Number of Bond : 4

Portfolio Duration : 1 year Number of Calculation : 10 (1 -10)

▶ Please enter your trading position data:
Only registered corporate bonds are allowed (you may Download a sample excel file for formatting).

Browse... Upload

Delete Submit

<input type="checkbox"/>	Order	Symbol	Error Status	Edit
<input type="checkbox"/>	1	AIS063A		Edit
<input type="checkbox"/>	2	LH06DA		Edit
<input type="checkbox"/>	3	PLC057A		Edit
<input type="checkbox"/>	4	BP064A		Edit
<input type="checkbox"/>	5	CPF063A		Edit

Delete Submit

รูป 7-1 แสดงรายละเอียดของการใส่ค่าอินพุทของแอปพลิเคชัน

จากรูป 7-1 จะเห็นได้ว่าอินพุทที่ทำการใส่เข้ามาคือ

1. หุ้นที่มีความสนใจอยู่ จากตัวอย่างนั้นจะมีการใส่หุ้นเข้ามาจำนวน 5 ตัว คือ AIS063A , LH06DA , PLC057A , BP064A และ CPF063A
2. จำนวนเงินที่ใช้ในการลงทุนกับพอร์ตนี้ (portfolio Value) จากตัวอย่างนั้นจำนวนเงินที่ใส่เข้ามา นั้นคือ 1,000,000
3. จำนวนของหุ้นที่ต้องการให้แอปพลิเคชันนี้ทำการเลือกให้ จากตัวอย่างนั้นหุ้นที่สนใจมีอยู่ 5 ตัว แต่ต้องการให้เลือกให้จำนวน 4 ตัว (Number of Bond)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ระยะเวลาที่จะใช้ในการลงทุนกับพอร์ตนี้ (Portfolio Duration) จากตัวอย่างต้องการลงทุนกับพอร์ตนี้เป็นระยะเวลา 1 ปี
5. จำนวนครั้งในการคำนวณซึ่งจะมีค่าได้ตั้งแต่ 1 ถึง 10 โดยจากตัวอย่างนี้กำหนดให้มีการคำนวณทั้งหมด 10 ครั้ง

เอาต์พุตของแอปพลิเคชันจะแบ่งออกเป็นสี่ส่วน โดยในส่วนที่หนึ่งนั้นจะแสดงรายละเอียดของหุ้นกู้ที่ได้จากอินพุต โดยจะแสดง ชื่อของหุ้นกู้ทุกตัวที่รับเข้ามาจากอินพุต (symbol) ,ราคาของหุ้นกู้ในแต่ละตัว (Price) ,ระยะเวลาที่เหลืออยู่ของหุ้นกู้ในแต่ละตัว (Duration) และค่าของคอนเวกซ์ซิตีของหุ้นกู้ในแต่ละตัว (Convexity) ซึ่งเอาต์พุตที่ได้นั้นเป็นไปตามรูป 7-2

Symbol	Price	Duration	Convexity
AIS063A	647.623496	1.057623	2467.942346
LH06DA	849.172094	1.479457	5931.641739
PLC057A	605.559121	0.632477	1180.968991
BP064A	848.481955	1.325284	4446.429408
CPF063A	868.23005	1.060537	3323.952812

รูป 7-2 แสดงเอาต์พุตที่เป็นรายละเอียดของหุ้นกู้ที่นักลงทุนสนใจ

ส่วนที่สองเป็นส่วนที่แสดงรายละเอียดของหุ้นกู้ที่แอปพลิเคชันเลือกในการคำนวณในแต่ละครั้ง โดยรายละเอียดในการคำนวณในแต่ละครั้งนั้นมีตามรูป 7-3 คือ

1. แสดงรายละเอียดของหุ้นกู้ที่โปรแกรมได้ทำการเลือก (5 เลือกเอามา 4)
2. จำนวนยูนิตของหุ้นกู้ในแต่ละตัวที่จะทำการซื้อ (Number of bond)
3. ราคาของหุ้นกู้ในแต่ละตัวคูณกับจำนวนยูนิตที่จะทำการซื้อ ได้ออกมาเป็นราคารวมของหุ้นกู้ในแต่ละตัว
4. ระยะเวลาที่ใช้ถือครองหุ้นกู้ในแต่ละตัว (Duration)
5. ค่าของคอนเวกซ์ซิตี (Convexity) ของหุ้นกู้ในแต่ละตัว

Calculation 1				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	616.098709	399000	0.421992	984.708996
LH06DA	77.722761	66000	0.097644	391.488355
PLC057A	515.226324	312000	0.197333	368.462325
BP064A	262.822325	223000	0.295538	991.553758
Total Value	1471.870119	1000000	1.012507	2736.213434

Calculation 2				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	457.055684	296000	0.313056	730.510935
LH06DA	14.131411	12000	0.017753	71.179701
PLC057A	582.932347	353000	0.223264	416.882054
BP064A	399.537077	339000	0.449271	1507.339569
Total Value	1453.656519	1000000	1.003344	2725.912259

Calculation 3				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	447.791042	290000	0.306711	715.70328
LH06DA	654.755383	556000	0.822578	3297.992807
PLC057A	247.704963	150000	0.094871	177.145349
BP064A	4.714302	4000	0.005301	17.785718
Total Value	1354.96569	1000000	1.229461	4208.627154

Calculation 4				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	665.510135	431000	0.455836	1063.683151
LH06DA	228.457614	194000	0.287015	1150.738497
PLC057A	582.932347	353000	0.223264	416.882054
BP064A	25.92866	22000	0.029156	97.821447
Total Value	1502.828956	1000000	0.995271	2729.125149

Calculation 5				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	375.218011	243000	0.257002	599.70999
LH06DA	281.450605	239000	0.35359	1417.662376
PLC057A	452.4744	274000	0.173299	323.585504
BP064A	287.572409	244000	0.323369	1084.928776
Total Value	1396.715425	1000000	1.10726	3425.886646

Calculation 6				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	1335.652591	865000	0.914844	2134.77013
LH06DA	9.420941	8000	0.011836	47.453134
PLC057A	203.11807	123000	0.077795	145.259186
BP064A	4.714302	4000	0.005301	17.785718
Total Value	1552.905904	1000000	1.009776	2345.268168

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Calculation 7				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	444.702828	288000	0.304595	710.767396
LH06DA	302.647722	257000	0.380221	1524.431927
PLC057A	683.665699	414000	0.261845	488.921162
BP064A	48.321593	41000	0.054337	182.303606
Total Value	1479.337842	1000000	1.000998	2906.424091

Calculation 8				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	277.939267	180000	0.190372	444.229622
LH06DA	467.514185	397000	0.587345	2354.861771
PLC057A	454.125766	275000	0.173931	324.766473
BP064A	174.429166	148000	0.196142	658.071552
Total Value	1374.008384	1000000	1.14779	3781.929418

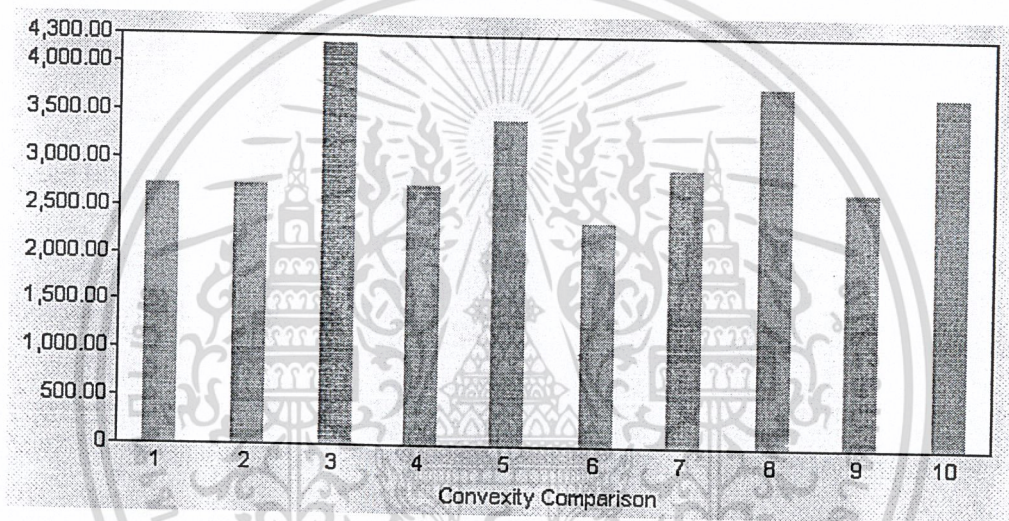
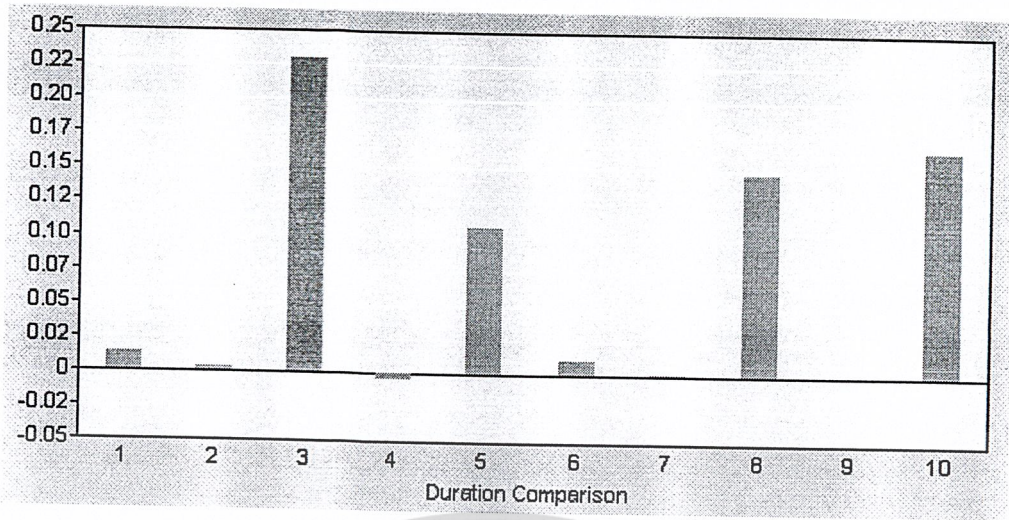
Calculation 9				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	514.187645	333000	0.352188	821.824801
LH06DA	11.776176	10000	0.014795	59.316417
PLC057A	566.418683	343000	0.216939	405.072364
BP064A	370.072691	314000	0.416139	1396.178834
Total Value	1462.455195	1000000	1.000061	2682.392416

Calculation 10				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	423.085329	274000	0.289789	676.216203
LH06DA	315.601516	268000	0.396495	1589.679986
PLC057A	305.502788	185000	0.117008	218.479263
BP064A	321.751097	273000	0.361803	1213.875229
Total Value	1365.94073	1000000	1.165095	3698.250681

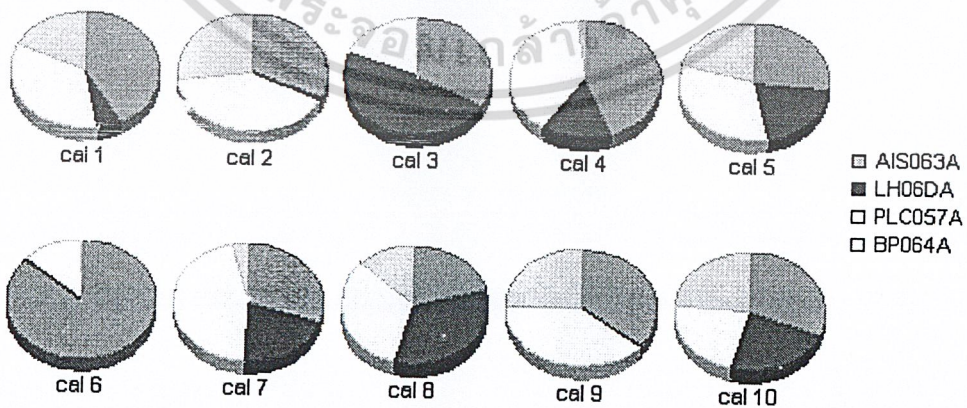
รูป 7-3 แสดงรายละเอียดของการคำนวณในแต่ละครั้ง

ส่วนที่สามแสดงรายละเอียดเป็นกราฟแห่ง 2 กราฟ โดยกราฟแรกนั้นแสดงค่าของความผิดพลาดที่เกิดขึ้นจากการคำนวณค่าของระยะเวลาที่ใช้ในการถือครองพอร์ต (Duration) ในแต่ละครั้ง ซึ่งค่าความผิดพลาดที่แสดงออกมานั้นได้ถูกนำมาคิดเป็นเปอร์เซ็นต์ เพื่อแสดงให้เห็นถึงความแตกต่างได้อย่างชัดเจน และในส่วนที่สองนั้นเป็นกราฟที่แสดงค่าของคอนเวกซ์ตี (Convexity) รวมในการคำนวณในแต่ละครั้ง ซึ่งจะเห็นได้จากรูป 7-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 7-4 แสดงรายละเอียดของค่าความแตกต่างของระยะเวลาในการถือครองหุ้นกู้ และค่าของคอนเวกซ์ตี
รวมใน การคำนวณในแต่ละครั้ง



รูป 7-5 แสดงรายละเอียดของสัดส่วนของหุ้นกู้ภายในพอร์ตในการคำนวณแต่ละครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนสุดท้าย (ส่วนที่สี่) แสดงออกมาในรูปแบบของกราฟแผนภูมิวงกลม (pie chart) ซึ่งจะบ่งบอกถึงสัดส่วนของหุ้นกู้ที่ทำการถือครองภายในพอร์ตในการคำนวณในแต่ละครั้ง

ตัวอย่างที่ 2

Calculation 1				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	174.175274	112800	0.29825	695.959742
LH06DA	137.545735	116800	0.432002	1732.039388
PLC057A	185.613586	112400	0.177726	331.852286
BP064A	69.357376	58000	0.192166	644.732264
Total Value	565.691971	400000	1.100144	3404.58368

Calculation 2				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	207.527986	134400	0.355361	829.228628
LH06DA	118.232807	100400	0.371344	1488.842077
PLC057A	165.797189	100400	0.158752	296.423217
BP064A	76.371689	64800	0.214696	720.321564
Total Value	567.929671	400000	1.100153	3334.815486

Calculation 3				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	171.08706	110800	0.292962	683.62003
LH06DA	123.885371	105200	0.389097	1560.021777
PLC057A	182.9714	110800	0.175196	327.128411
BP064A	86.271723	73200	0.242527	813.696582
Total Value	564.215554	400000	1.099782	3384.4668

Calculation 4				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	178.498774	115600	0.305653	713.235338
LH06DA	159.684946	135600	0.501536	2010.82655
PLC057A	190.897958	115600	0.182786	341.300038
BP064A	39.128705	33200	0.109999	369.053641
Total Value	568.210383	400000	1.099974	3434.415567

Calculation 5				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	123.528563	80000	0.211525	493.588469
LH06DA	96.564643	82000	0.303289	1215.986557
PLC057A	194.200691	117600	0.185948	347.204883
BP064A	141.900484	120400	0.398911	1338.375252
Total Value	556.194381	400000	1.099673	3395.155161

Calculation 6				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	179.116417	116000	0.306711	715.70328
LH06DA	152.148193	129200	0.477865	1915.920282
PLC057A	188.255772	114000	0.180256	336.576162
BP064A	48.085878	40800	0.135179	453.5358
Total Value	567.60626	400000	1.100011	3421.735524

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

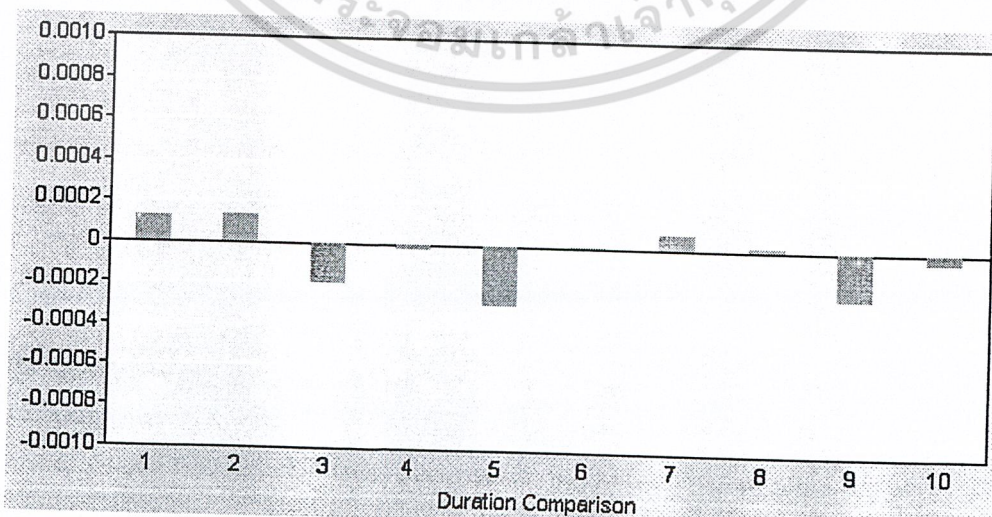
Calculation 7				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	235.939556	152800	0.404012	942.753976
LH06DA	140.843064	119600	0.442358	1773.56088
PLC057A	161.173363	97600	0.154324	288.156434
BP064A	35.357263	30000	0.099396	333.482206
Total Value	573.313246	400000	1.10009	3337.953496

Calculation 8				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	208.763272	135200	0.357477	834.164513
LH06DA	159.684946	135600	0.501536	2010.82655
PLC057A	178.347574	108000	0.170769	318.861628
BP064A	24.985799	21200	0.07024	235.660759
Total Value	571.781591	400000	1.100022	3399.51345

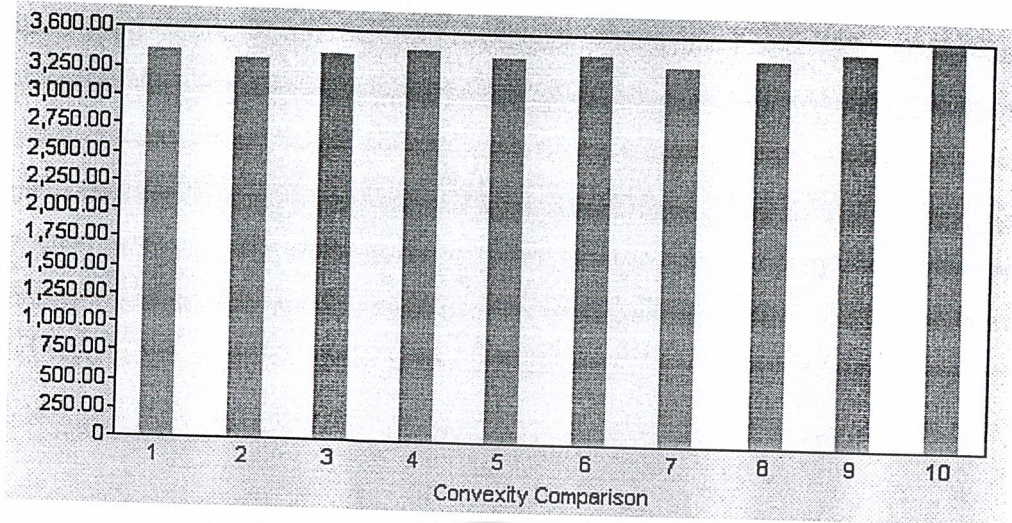
Calculation 9				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	185.292845	120000	0.317287	740.382704
LH06DA	187.005674	158800	0.587345	2354.861771
PLC057A	196.842878	119200	0.188478	351.928759
BP064A	2.357151	2000	0.006626	22.232147
Total Value	571.498548	400000	1.099736	3469.405381

Calculation 10				
Symbol	Number of Bond	Total Price	Duration	Convexity
AIS063A	25.940998	16800	0.04442	103.653579
LH06DA	137.545735	116800	0.432002	1732.039388
PLC057A	247.044417	149600	0.236546	441.682403
BP064A	137.657612	116800	0.386983	1298.357387
Total Value	548.188762	400000	1.099951	3575.732757

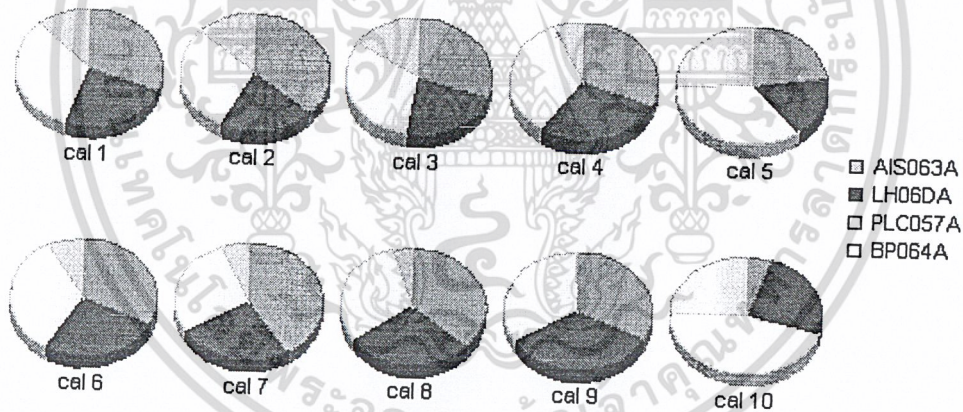
รูป 7-6 แสดงรายละเอียดของการคำนวณในแต่ละครั้ง โดยระยะเวลาในการถือครองพอร์ตมีค่าเท่ากับ 1.1 ปี และมีการใช้เทคนิคการรีโปรดักชันในการคำนวณ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 7-7 แสดงรายละเอียดของค่าความแตกต่างของระยะเวลาในการถือครองหุ้นกู้และค่าของคอนเวกซ์ตีรวมใน การคำนวณในแต่ละครั้ง และมีการใช้เทคนิคการรีโปรดักชันในการคำนวณ



รูป 7-8 แสดงรายละเอียดของสัดส่วนของหุ้นกู้ภายในพอร์ตในการคำนวณแต่ละครั้ง และมีการใช้เทคนิคการรีโปรดักชันในการคำนวณ

ตัวอย่างที่ 1 นั้นได้กำหนดจำนวนรุ่น (Generation) เอาไว้ที่ 80 รุ่น และ ไม่ได้มีการใช้เทคนิคของการรีโปรดักชันเลย ซึ่งจะเห็นได้ว่า ความผิดพลาดที่เกิดขึ้นในส่วนของระยะเวลาในการถือครองพอร์ต (Duration) นั้นค่อนข้างมีความแตกต่างกันในการคำนวณในแต่ละครั้ง และค่อนข้างมีความผิดพลาดมากเกินไป โดยในตัวอย่างที่ 2 นั้น จะทำการเพิ่มจำนวนรุ่น (Generation) เป็น 120 รุ่น และทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้เทคนิคการรีโพรดักชัน แบบเก็บส่วนดีไว้(Reproducion) ในการคำนวณด้วย จากผลการทำงานของโปรแกรมตามตัวอย่างที่ 2 จะเห็นได้ว่าความผิดพลาดที่เกิดขึ้นนั้นมีน้อยมาก หรือแทบจะไม่มีเลยก็ว่าได้ ค่าของคอน-เวกซ์ตีในการคำนวณในแต่ละครั้งก็ค่อนข้างมีความใกล้เคียงกันมาก และหน่วยที่ถือครองหุ่นกู่ในแต่ละตัวในการคำนวณในแต่ละครั้งจะมีแนวโน้มไปในทิศทางเดียวกัน ในการทดลองเพิ่มจำนวนรูนให้มากกว่า 120 รูนนั้นผลที่ได้ออกมานั้นไม่ได้มีความแตกต่างจากการกำหนดจำนวนรูนเอาไว้ที่ 120 รูนเลย แต่จะทำให้ใช้เวลาในการทำงานเพิ่มจากเดิมขึ้นมาก ดังนั้นแอปพลิเคชันนี้จึงได้ทำการกำหนดค่าของจำนวนรูนเอาไว้ที่ 120 รูน และทำการกำหนดจำนวนประชากรในแต่ละรูนเอาไว้ที่ 10 โครโมโซมต่อหนึ่งรูน

จากการทดลองที่ผ่านมามีค่าที่เหมาะสมในการใช้งานกับแอปพลิเคชันนี้ คือ

- จำนวนประชากรในแต่ละรูนมีค่าเท่ากับ 10 โครโมโซมต่อหนึ่งรูน
- จำนวนรูนมีค่าเท่ากับ 120 รูน
- อัตราส่วนในการครอสโอเวอร์มีค่าเท่ากับ 0.7 และ
- อัตราส่วนในการมิวเตชันมีค่าเท่ากับ 0.01



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

แพ็คเกจ GeneticAlgorithms.GeneticStruct

```

Namespace GeneticAlgorithms.geneticStruct
{
public class Chromosome
{
    private double[] _UNITS;
    private double _dFITNESS;
    private double _cFITNESS;

    public Chromosome(int numOfGenes)
    {
        _UNITS = new double[numOfGenes];
    }

    public double[] UNITS
    {
        get {return _UNITS;}
        set {_UNITS = value;}
    }

    public double dFITNESS
    {
        get {return _dFITNESS;}
        set {_dFITNESS = value;}
    }

    public double cFITNESS
    {
        get {return _cFITNESS;}
        set {_cFITNESS = value;}
    }

    public Chromosome clone()
    {
        Chromosome newch = new Chromosome(this.UNITS.Length);
        for(int i = 0; i < this.UNITS.Length ; i ++)
            newch.UNITS[i] = this.UNITS[i];

        newch.dFITNESS = this.dFITNESS;
        newch.cFITNESS = this.cFITNESS;

        return newch;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพ็คเกจ GeneticAlgorithms.GeneticBiz

คลาส GeneticBiz

```

namespace GeneticAlgorithms.geneticBiz
{
public class GeneticBiz
{
private InputParams inputObj;
private BondDataList bondInput;

private int population = 10;
private int generation = 150;
private double crossoverRatio = 0.7;
private double mutationRatio = 0.01 ;

private ArrayList inipool = new ArrayList();
private ArrayList offspringPool = new ArrayList();
private ArrayList bestChrOfeachGen = new ArrayList();

private BondDataList bondComb;
private Combination indexComb;
private StructPort ansPort = new StructPort();

public GeneticBiz(){}

public void init(InputParams obj,BondDataList input)
{
inputObj = obj;
bondInput = input;

indexComb = new
Combination((double)bondInput.Count, (double)inputObj.NBOND);
}

public void geneticAlgorithms()
{
GeneticOp gaOp = new GeneticOp();
GeneticEqu gaEqu = new GeneticEqu();
Reproduction reproduct = new Reproduction();

inipool = new ArrayList();
offspringPool = new ArrayList();
bestChrOfeachGen = new ArrayList();
ansPort = new StructPort();

Random r = new Random();
double probab = 0;

ArrayList bonds = bondInput.getBondDataList();

for(int i = 0; i < Combination.Choose((double)
bondInput.Count, (double)inputObj.NBOND); i++)
{
bondComb = new BondDataList();
Combination tmpIndex = indexComb.Element(i);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double[] index = tmpIndex.getData();

//generate bond combinations
for(int count = 0 ; count < index.Length ; count++)
{
    BondData bond = (BondData) bonds[count];
    this.bondComb.append(bond);
}

//use genetic algorithms for its units

//find initial population randomly
creatInitialPopulation();
for(int gen = 0 ; gen < generation ; gen++)
{
    //find fitness for each chromosome in matingpool
    offspringPool.Clear();

    for(int j = 0; j < population ; j++)
        gaEqu.pmFitnessFunction(bondComb, (Chromosome)
            inipool[j],inputObj.DURATION);

    //build roulettewheel
    RWselection rw = new RWselection();
    rw.buildRW(inipool);

    //begin iteration process until offspring pool
    size is equal to population
    for(int pop = 0 ; pop < population/2 ; pop++)
    {
        //choose parent chromosome by RoulettWheel;
        double parent1_Index = rw.RWSelection();
        double parent2_Index = rw.RWSelection();

        Chromosome[] matingChr = new Chromosome[2];

        Chromosome tmpMate1 = (Chromosome)
            inipool[(int) parent1_Index];
        matingChr[0] = tmpMate1.clone();
        Chromosome tmpMate2 = (Chromosome)
            inipool[(int) parent2_Index];
        matingChr[1] = tmpMate2.clone();

        //use crossover operator
        prob = r.NextDouble();

        if(prob < crossoverRatio)
            matingChr = gaOp.pmCrossOver(matingChr);

        offspringPool.Add(matingChr[0]);
        offspringPool.Add(matingChr[1]);
    }

    //use mutation
    for(int pop = 0 ; pop < population ; pop++)
    {
        prob = r.NextDouble();

        if(prob < mutationRatio)

            gaOp.pmMutation((Chromosome)
                offspringPool[pop]);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    //use reproduction process to keep the best
    chromosome

    inipool =
    reproduct.getGeneration(inipool,offspringPool);
    //offspringPool.Clear();

    bestChrOfeachGen.Add(reproduct.BESTCHR_PREVGEN);

    }//end all generation

    //use objective function to find best chromosome for
    this combination
    Chromosome bestChr =
    gaEqu.pmObjectiveFunction(bestChrOfeachGen);

    compareBestCombination(bestChr);

    }//end combination
}

private void compareBestCombination(Chromosome bestch)
{
    if(bestch.dFITNESS > ansPort.dFITNESS)
    {
        //set bond datalist
        ArrayList bonds = bondComb.getBondDataList();
        BondDataList tmpList = new BondDataList();
        for(int i = 0 ; i < bonds.Count ; i++)
        {
            tmpList.append((BondData) bonds[i]);
        }
        ansPort.BONDS = tmpList;

        //set duration fitness
        ansPort.dFITNESS = bestch.dFITNESS;

        //set convexity fitness
        ansPort.cFITNESS = bestch.cFITNESS;

        //set units
        double[] tmpUnit = new double[bestch.UNITS.Length];
        for(int j = 0 ; j < bestch.UNITS.Length ; j ++ )
        {
            tmpUnit[j] = bestch.UNITS[j];
        }
        ansPort.UNITS = tmpUnit;
    }
}

private void creatInitialPopulation()
{
    inipool.Clear();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double sum = 0;
double sumUnit = 0;
double randomValue = 0;

Random r = new Random();

for(int i = 0 ; i < population ; i++)
{
    sum = 0;
    Chromosome chr = new Chromosome(inputObj.NBOND);

    for(int j = 0 ; j <    inputObj.NBOND ; j++)
    {
        randomValue = 0;
        while(randomValue == 0)
        {
            randomValue = r.Next(100);
        }
        chr.UNITS[j] = randomValue;
        sum += chr.UNITS[j];
    }

    sumUnit = 0;
    for(int j = 0 ; j <    inputObj.NBOND ; j++)
    {
        if(j != inputObj.NBOND-1)
        {
            chr.UNITS[j] =
            Math.Round((chr.UNITS[j]/sum),3);
            sumUnit += chr.UNITS[j];
        }
        else
        {
            chr.UNITS[j] = Math.Round((1.0 - sumUnit),3);
        }
    }
    inipool.Add(chr);
}
}

public StructPort getAnsPort()
{
    return ansPort;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส Combination

```

namespace GeneticAlgorithms.geneticBiz
{
    public class Combination
    {
        private double n = 0;
        private double k = 0;
        private double[] data = null;

        public Combination(double n, double k)
        {
            if (n < 0 || k < 0) // normally require n >= k
                throw new Exception("Negative parameter in
                    constructor");

            this.n = n;
            this.k = k;
            this.data = new double[(int)k];
            for (int i = 0; i < k; ++i)
                this.data[i] = i;
        } // Combination(n,k)

        public Combination(double n, double k, double[] a) //
            Combination from a[]
        {
            if (k != a.Length)
                throw new Exception("Array length does not
                    equal k");
            this.n = n;
            this.k = k;
            this.data = new double[(int)k];
            for (int i = 0; i < a.Length; ++i)
                this.data[i] = a[i];

            if (!this.IsValid())
                throw new Exception("Bad value from array");
        } // Combination(n,k,a)

        public bool IsValid()
        {
            if (this.data.Length != this.k)
                return false; // corrupted

            for (int i = 0; i < this.k; ++i)
            {
                if (this.data[i] < 0 || this.data[i] >
                    this.n - 1)
                    return false; // value out of range

                for (int j = i+1; j < this.k; ++j)
                    if (this.data[i] >= this.data[j])
                        return false; // duplicate
                        or not lexicographic
            }

            return true;
        } // IsValid()
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public override string ToString()
{
    string s = "{ ";
    for (int i = 0; i < this.k; ++i)
        s += this.data[i].ToString() + " ";
    s += "}";
    return s;
} // ToString()

public Combination Successor()
{
    if (this.data[0] == this.n - this.k)
        return null;

    Combination ans = new Combination(this.n, this.k);

    int i;
    for (i = 0; i < this.k; ++i)
        ans.data[i] = this.data[i];

    for (i = (int)this.k - 1; i > 0 && ans.data[i] ==
        this.n - this.k + i; --i)
        ;
    ++ans.data[i];

    for (int j = i; j < this.k - 1; ++j)
        ans.data[j+1] = ans.data[j] + 1;

    return ans;
} // Successor()

public Combination First()
{
    Combination ans = new Combination(this.n, this.k);

    for (int i = 0; i < ans.k; ++i)
        ans.data[i] = i;

    return ans;
} // First()

public string[] ApplyTo(string[] strarr)
{
    if (strarr.Length != this.n)
        throw new Exception("Array size not equal to
        Combination order in ApplyTo()");

    string[] result = new string[(int)this.k];

    for (int i = 0; i < result.Length; ++i)
        result[i] = strarr[(int)this.data[i]];

    return result;
} // ApplyTo()

public static double Choose(double n, double k)
{
    if (n < 0 || k < 0)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        throw new Exception("Invalid negative
        parameter in Choose()");
    if (n < k)
        return 0; // special case
    if (n == k)
        return 1;

    double delta, iMax;

    if (k < n-k) // ex: Choose(100,3)
    {
        delta = n-k;
        iMax = k;
    }
    else // ex: Choose(100,97)
    {
        delta = k;
        iMax = n-k;
    }

    double ans = delta + 1;
    for (int i = 2; i <= iMax; ++i)
    {
        checked { ans = (ans * (delta + i)) / i; }
        // Throws Overflow Exception
        //Console.WriteLine(" i = " + i + " n = " + n + "
        k = " + k + " delta = " + delta + " iMax = " +
        iMax + " ans = " + ans);
    }
    return ans;
} // Choose()

// return the mth lexicographic element of combination
C(n,k)
public Combination Element(double m)
{
    double[] ans = new double[(int)this.k];
    double a = this.n;
    double b = this.k;
    double x = (Choose(this.n, this.k) - 1) - m; // x
    is the "dual" of m

    for (int i = 0; i < this.k; ++i)
    {
        ans[i] = LargestV(a,b,x); // largest value
        v, where v < a and vCb < x
        x = x - Choose(ans[i],b);
        a = ans[i];
        b = b-1;
    }

    for (int i = 0; i < this.k; ++i)
    {
        ans[i] = (n-1) - ans[i];
    }

    return new Combination(this.n, this.k, ans);
} // Element()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public double[] getData()
{
    return this.data;
}

// return largest value v where v < a and Choose(v,b)
<= x
private static double LargestV(double a, double b,
double x)
{
    double v = a - 1;
    while (Choose(v,b) > x)
        --v;
    return v;
} // LargestV()

} // class Combination
} // ns CombinationsLib

```

คลาส GeneticAlgorithms.GeneticEquation

```

namespace GeneticAlgorithms.geneticEquation
{
    public class GeneticEqu
    {
        public GeneticEqu()
        {}
        public void pmFitnessFunction(BondDataList
bonds, Chromosome chr, double ExpDuration)
        {
            double calculateDuration = 0;
            double calculateConvexity = 0;
            ArrayList tempBond = bonds.getBondDataList();
            for(int i = 0; i < tempBond.Count; i++)
            {
                BondData calBond = (BondData)tempBond[i];
                //--calculate Duration--//
                calculateDuration +=
                chr.UNITS[i]*calBond.TTM;
                //--calculate Convexity--//
                calculateConvexity +=
                chr.UNITS[i]*calBond.TotalPVBP;
            }
            double diff = Math.Abs(ExpDuration -
            calculateDuration);
            chr.cFITNESS = calculateConvexity;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if(diff != 0)
    {
        chr.dFITNESS = Math.Round(1/diff, 6);
    }
    else
    {
        chr.dFITNESS = 1/0.000001;
    }
}

public Chromosome pmObjectiveFunction(ArrayList Gen)
{
    Chromosome chr = (Chromosome)Gen[0];
    Chromosome tmpChr = new
    Chromosome(chr.UNITS.Length);

    for(int i = 0; i < Gen.Count; i++)
    {
        Chromosome chr0 = (Chromosome)Gen[i];
        for(int j = 0; j < Gen.Count; j++)
        {
            Chromosome chr1 = (Chromosome)Gen[j];
            if(chr0.dFITNESS > chr1.dFITNESS)
            {
                tmpChr = (Chromosome)Gen[i];
                Gen[i] = Gen[j];
                Gen[j] = tmpChr;
            }
            else if(chr0.dFITNESS ==
            chr1.dFITNESS)
            {
                if(chr0.cFITNESS > chr.cFITNESS)
                {
                    tmpChr = (Chromosome)Gen[i];
                    Gen[i] = Gen[j];
                    Gen[j] = tmpChr;
                }
            }
        }
    }
    return (Chromosome)Gen[0];
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพ็คเกจ GeneticAlgorithms.GeneticOperator

คลาส GeneticOperator

```

namespace GeneticAlgorithms.geneticOperator
{
public class GeneticOp
{
    public GeneticOp()
    {}

    private void rescaleUnit(Chromosome chr,double diff,int point)
    {
        Chromosome tmpChr = chr.clone();
        bool negative = false;

        double avgUnit = diff/(chr.UNITS.Length-1.0);
        for(int i = 0 ; i < tmpChr.UNITS.Length ; i++)
        {
            if(i != point)
            {
                tmpChr.UNITS[i] += Math.Round(avgUnit,3);
                tmpChr.UNITS[i] = Math.Round(tmpChr.UNITS[i]);
                if(tmpChr.UNITS[i] <= 0)
                    negative = true;
            }
        }
        if(negative)
        {
            double sum = 0;
            for(int i = 0 ; i < chr.UNITS.Length ; i++)
            {
                if(point != chr.UNITS.Length-1)
                {
                    if(i != point)
                    {
                        if(i != chr.UNITS.Length-1)
                        {
                            avgUnit = ((1.0-chr.UNITS[point])*
                                chr.UNITS[i]) / (1.0-
                                    (chr.UNITS[point]+diff));

                            chr.UNITS[i] = Math.Round(avgUnit,3);

                            sum += chr.UNITS[i];
                        }
                        else
                        {
                            chr.UNITS[i] = Math.Round((1.0 -
                                sum - chr.UNITS[point]),3);
                            chr.UNITS[point] =
                                Math.Round(chr.UNITS[point],3);
                        }
                    }
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        }
    }
    }
    else
    {
        if(i != point)
        {
            if(i != chr.UNITS.Length-2)
            {
                avgUnit = ((1.0- chr.UNITS[point])* chr.UNITS[i])
                /(1.0-(chr.UNITS[point]+diff));
                chr.UNITS[i] = Math.Round(avgUnit,3);
                sum += chr.UNITS[i];
            }
            else
            {
                chr.UNITS[i] = Math.Round((1.0 - sum -
                chr.UNITS[point]),3);
                chr.UNITS[point] =
                Math.Round(chr.UNITS[point],3);
            }
        }
    }
}
else
chr = tmpChr;
}
public Chromosome[] pmCrossOver(Chromosome[] matingChrs)
{
    int[] crossPoint = new int[matingChrs.Length];
    double[] diff = new double[matingChrs.Length];
    double tempUnit = 0;

    //random for crossover positions
    Random r = new Random();

    crossPoint[0] = r.Next(0,matingChrs[0].UNITS.Length);
    crossPoint[1] = r.Next(0,matingChrs[1].UNITS.Length);

    //find different between old and new unit
    diff[0] = matingChrs[0].UNITS[crossPoint[0]] -
    matingChrs[1].UNITS[crossPoint[1]];
    diff[1] = matingChrs[1].UNITS[crossPoint[1]] -
    matingChrs[0].UNITS[crossPoint[0]];

    //crossover operation
    tempUnit = matingChrs[1].UNITS[crossPoint[1]];
    matingChrs[1].UNITS[crossPoint[1]] =
    matingChrs[0].UNITS[crossPoint[0]];
    matingChrs[0].UNITS[crossPoint[0]] = tempUnit;

    for(int i = 0 ; i < matingChrs.Length ; i++)
        rescaleUnit(matingChrs[i],Math.Round(diff[i],3),crossPoint[i])

    return matingChrs;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

public void pmMutation(Chromosome muteChr)
{
    int mutePoint = 0;
    double diff = 0;
    double muteUnit = 0;

    //random for mutation point and mutation unit
    Random r = new Random();

    mutePoint = r.Next(0,muteChr.UNITS.Length);
    muteUnit = r.NextDouble();

    //find different
    diff = muteChr.UNITS[mutePoint] - muteUnit;

    //mutate
    muteChr.UNITS[mutePoint] = muteUnit;

    rescaleUnit(muteChr,Math.Round(diff,3),mutePoint);
}
}
}

```

คลาส RWselection

```

namespace GeneticAlgorithms.geneticOperator
{
    public class RWselection
    {
        private DataTable rwTable = new DataTable();
        public RWselection()
        {
            rwTable.Columns.Add(new
            DataColumn("L_Bound",typeof(double)));
            rwTable.Columns.Add(new
            DataColumn("U_Bound",typeof(double)));
            rwTable.Columns.Add(new
            DataColumn("CH_Inx",typeof(int)));
        }

        public void buildRW(ArrayList gtChromosome)
        {
            double begin = 0;
            double sum = 0;

            for(int i = 0; i < gtChromosome.Count; i++)
            {
                Chromosome chr =
                (Chromosome)gtChromosome[i];

                sum += chr.dFITNESS;
            }

            for(int i = 0; i < gtChromosome.Count; i++)
            {
                Chromosome ch = (Chromosome)gtChromosome[i];
                double area = (100/sum)*ch.dFITNESS;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        DataRow dr = rwTable.NewRow();

        dr["L_Bound"] = begin;
        dr["U_Bound"] = begin + area;
        dr["CH_Inx"] = i;

        begin += area;

        rwTable.Rows.Add(dr);

    }

}

public double RWSelection()
{
    double rwValue = 0;

    Random rwRandom = new Random();
    rwValue = rwRandom.Next(100);

    for(int i = 0 ; i < rwTable.Rows.Count ; i++)
    {
        if(rwValue == 0.0)
            return 0;

        if(Convert.ToDouble(rwTable.Rows[i]["L_Bound"]) <
            rwValue &&
            Convert.ToDouble(rwTable.Rows[i]["U_Bound"]) >=
            rwValue)
            return i;
    }

    return 0;
}
}
}

```

กลาส Reproduction

```

namespace GeneticAlgorithms.geneticOperator
{
    public class Reproduction
    {
        private Chromosome bestChr_prevGen;

        public Reproduction()
        {}

        public Chromosome BESTCHR_PREVGEN
        {
            get {return bestChr_prevGen;}
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        set {bestChr_prevGen = value;}
    }

public ArrayList getGeneration(ArrayList G1, ArrayList G2)
{
    Chromosome chr = (Chromosome)G1[0];
    Chromosome tmpChr = new Chromosome(chr.UNITS.Length);

    for(int i = 0; i < G1.Count; i++)
    {
        Chromosome G1_chr0 = (Chromosome)G1[i];
        Chromosome G2_chr0 = (Chromosome)G2[i];

        for(int j = 0; j < G1.Count; j++)
        {
            Chromosome G1_chr1 = (Chromosome)G1[j];
            Chromosome G2_chr1 = (Chromosome)G2[j];

            if(G1_chr0.dFITNESS > G1_chr1.dFITNESS)
            {
                tmpChr = (Chromosome)G1[i];
                G1[i] = G1[j];
                G1[j] = tmpChr;
            }
            if(G2_chr0.dFITNESS < G2_chr1.dFITNESS)
            {
                tmpChr = (Chromosome)G2[i];
                G2[i] = G2[j];
                G2[j] = tmpChr;
            }
        }
        Chromosome best = (Chromosome)G1[0];
        bestChr_prevGen = best.clone();
        tmpChr = (Chromosome)G1[0];
        G1[0] = G2[0];
        G2[0] = tmpChr;

        ArrayList GenNext = new ArrayList();
        Generation gen = new Generation(G2);
        GenNext = gen.clone();

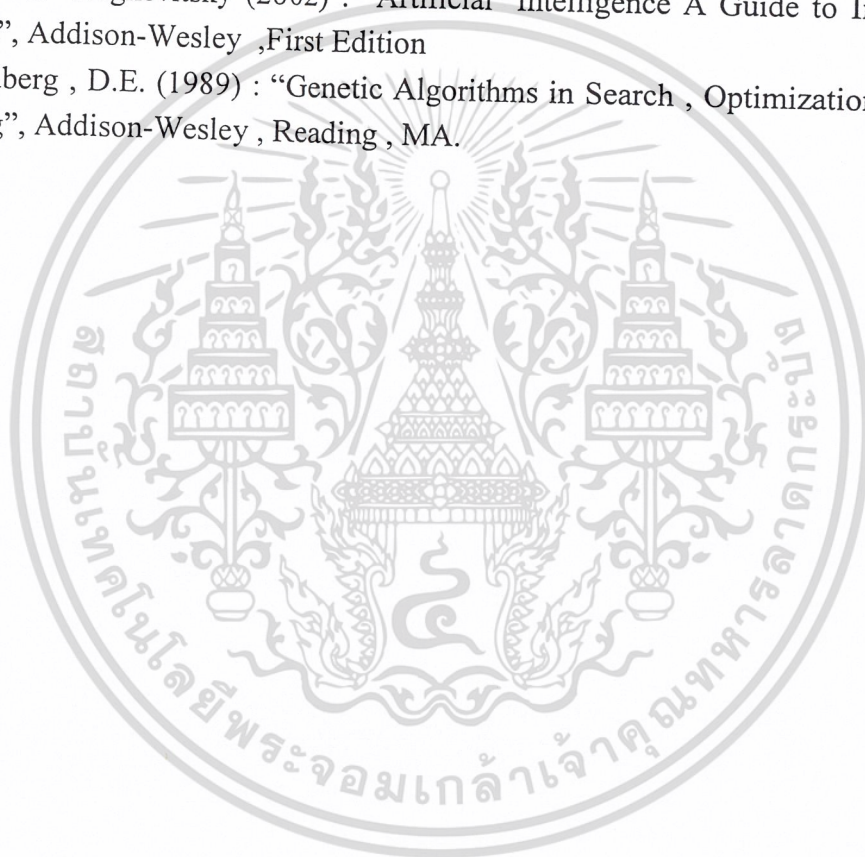
        return GenNext;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] ศูนย์ซื้อขายตราสารหนี้ไทย : “ก้าวสู่การลงทุนในตลาดตราสารหนี้”, อินฟอร์มีเดียร์บุ๊ก 2545
- [2] Holland ,J.H. (1975) : “Adaptation in Natural and Artificial Systems , University of Michigan Press” , Ann Arbor
- [3] Turing , A.M. (1950) : “Computation Machinery and Intelligence , Mind,59,433-460
- [4] Mitchell , M (1996): “An Introduction to Genetic Algorithms”, MIT Press ,Cambridge ,MA
- [5] Michael Negnevitsky (2002) : “Artificial Intelligence A Guide to Intelligence Systems”, Addison-Wesley ,First Edition
- [6] Goldberg , D.E. (1989) : “Genetic Algorithms in Search , Optimization Machine Learning”, Addison-Wesley , Reading , MA.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้