บทที่ 6

ฟัซซี่เจนเนติกอัลกอริทึมสำหรับปัญหาการจัดลำดับงานเข้าสายการประกอบแบบผลิตภัณฑ์ผสมที่มีเวลาการทำงานเป็นฟัซซี่

เนื้อหาในบทนี้กล่าวถึงการประยุกต์ใช้ฟัซซี่เจนเนติกอัลกอริทึมสำหรับปัญหาการจัดลำดับงานเข้าสายการประกอบแบบผลิตภัณฑ์ผสมที่มีเวลาการทำงานเป็นฟัซซี่ โดยกล่าวถึงลักษณะของปัญหา โครงสร้างของเจนเนติกอัลกอริทึมขั้นตอนและวิธีการของเจนเนติกอัลกอริทึม สำหรับปัญหาที่ทำการศึกษาในงานวิจัย

6.1 ลักษณะของปัญหาการจัดลำดับผลิตภัณฑ์เข้าสายการประกอบแบบผลิตภัณฑ์ผสมที่มีเวลาการทำงานเป็นแบบฟัซซี่

ปัญหาการจัดลำดับงานเข้าสายการประกอบแบบผลิตภัณฑ์ผสมทีมีเวลาการทำงานเป็นแบบฟัซซี่ มีลักษณะดังนี้

6.1.1 ปัญหาการจัดลำดับผลิตภัณฑ์ที่จะเข้าทำการประกอบในสายการประกอบผลิตภัณฑ์ผสม คือ มีการประกอบผลิตภัณฑ์ 2 ชนิด หรือมากกว่า โดยผลิตภัณฑ์ต่างๆ จะเข้าสู่สายงานการประกอบปะปนกัน ไม่มีการแบ่งว่าต้องทำผลิตภัณฑ์ชุดไหนก่อน

6.1.2 เป็นสายการประกอบที่มีระบบการผลิตแบบลื่นไหล ประกอบด้วยสถานีงานต่างๆ โดยกำหนดจำนวนสถานีงานและขั้นงานในแต่ละสถานีงาน และสายการประกอบเป็นอนุกรม คือ สถานีงานทำงานต่อเนื่องกันตามลำดับ ไม่มีสถานีงานใดทำงานขนานกัน

6.1.3 การจัดลำดับผลิตภัณฑ์จะจัดเข้าเป็นกลุ่ม โดยใช้สัดส่วนของผลิตภัณฑ์ที่ทำการผลิต (Minimum Part Set :MPS) ซึ่งเป็นการจัดลำดับที่ตอบสนองความต้องการด้านความหลากหลายของรูปแบบผลิตภัณฑ์อย่างเหมาะสม

6.1.4 ข้อมูลเข้า (Input) คือ

- ชนิดและจำนวนผลิตภัณฑ์
- ขั้นการทำงานของแต่ละผลิตภัณฑ์ (Work Element)
- จำนวนสถานีงานและขั้นงานในสถานีงานต่างๆ
- เวลาทำงานในแต่ละขั้นงาน (Processing Time)

6.1.5 ทุกสถานีงานมีเวลาในการทำงานเท่ากัน

6.1.6 เวลาการทำงานในแต่ละขั้นงาน มีลักษณะแบบฟัซซี่ โดยมีลักษณะฟังก์ชันการเป็นสมาชิกเป็นรูปสามเหลี่ยม (Triangular Fuzzy) ประกอบด้วย เวลาใช้ที่ต่ำสุด เวลาที่ใช้เฉลี่ย และเวลาที่ใช้สูงสุด โดยสามารถแสดงค่าเวลาการทำงานในรูปของฟังก์ชันการเป็นสมาชิก (Membership Function) ของเซตฟัซซี่

6.1.7 เวลาการทำงานของแต่ละงานในแต่ละสถานี จะทำการรวมตามวิธีการของฟัซซี่เซต (Addition of Fuzzy) โดยการนำค่าเวลาที่ใช้ต่ำที่สุด เวลาที่ใช้เฉลี่ย และเวลาที่ใช้สูงสุดของแต่ละขั้นงานมาบวกกัน

6.1.8 เวลาการทำงานเสร็จสิ้นจะพิจารณาจากเวลาที่งานสุดท้ายทำเสร็จ บนเครื่องจักรเครื่องสุดท้าย ซึ่งได้จากการรวมของเวลาการทำงานในแต่ละขั้นงานของแต่ละผลิตภัณฑ์และเวลาว่าง

6.1.9 การจัดลำดับผลิตภัณฑ์ที่เข้าทำการประกอบในสายงานการประกอบ ทำเพื่อตอบสนองวัตถุประสงค์ คือ เพื่อให้เวลาที่ใช้ในการประกอบทุกผลิตภัณฑ์เสร็จสิ้นน้อยที่สุด โดยพิจารณาจากค่าความพึงพอใจของผู้ทำการตัดสินใจ

คำตอบที่ได้เป็นคำตอบที่ตอบสนองค่าความพอใจของผู้ทำการตัดสินใจ โดยใช้ Max-Min Operator ในการแก้ปัญหา ของเซตฟัซซี่ 2 เซต ได้แก่ เซตของวัตถุประสงค์ (Objective Function) และเซตของเวลาการทำงานเสร็จสิ้น (Makespan Function)

6.1.10 ข้อกำหนดที่ต้องพิจารณาในการแก้ปัญหา มีดังนี้

- ผลิตภัณฑ์หลากหลายชนิดที่ทำการจัดลำดับเพื่อเข้าทำการประกอบจะต้องทำการประกอบบนสายการประกอบเดียวกัน
- ในการเข้าทำการประกอบในแต่ละสถานีงานจะทำตามลำดับของผลิตภัณฑ์ที่จัดไว้ ไม่มีการอนุญาตให้มีการแทรกงานอื่นเข้ามาในสถานีงานนั้นๆ เมื่อผลิตภัณฑ์ก่อนหน้ายังประกอบไม่เสร็จสิ้น

6.2 โครงสร้างของเจนเนติกอัลกอริทึม

โครงสร้างหลักของเจนเนติกอัลกอริทึม ประกอบด้วย 5 ส่วน คือ

1. **Initialization** เป็นการใส่รหัสคำตอบและสร้างสตริงคำตอบเบื้องต้น
2. **Reproduction** เป็นการคัดเลือกสตริงคำตอบที่มีความเหมาะสม โดยดูจากค่าความเหมาะสม (Fitness) ของสตริงคำตอบแต่ละตัว ในส่วนของการ Reproduction จะแบ่งเป็นขั้นตอนย่อย 3 ขั้นตอน คือ

2.1 Decoding เป็นการแปลความหมายของสตริงคำตอบ ในที่นี้คือการถอด รหัสคำตอบ

2.2 Evaluation เป็นการคำนวณหาค่าต่างๆเพื่อนำไปสู่การหาค่าความเหมาะสม (Fitness) ของสตริงคำตอบ

2.3 Selection เป็นกระบวนการในการคัดเลือกสตริงที่มีความเหมาะสมมากกว่า

3. **Crossover** เป็นการสร้างสตริงคำตอบตัวใหม่จากการแลกเปลี่ยนชิ้นส่วนระหว่างสตริงคำตอบ 2 ตัว
4. **Mutation** เป็นการสร้างสตริงคำตอบตัวใหม่โดยการย้ายค่าบางตำแหน่งภายในสตริงคำตอบ
5. **Elite Preserve Strategy** เป็นการเก็บค่าที่ดีที่สุดที่ได้จากการเปรียบเทียบระหว่างสตริงคำตอบที่สร้างใหม่ กับสตริงคำตอบตัวที่ดีที่สุดตัวเดิม

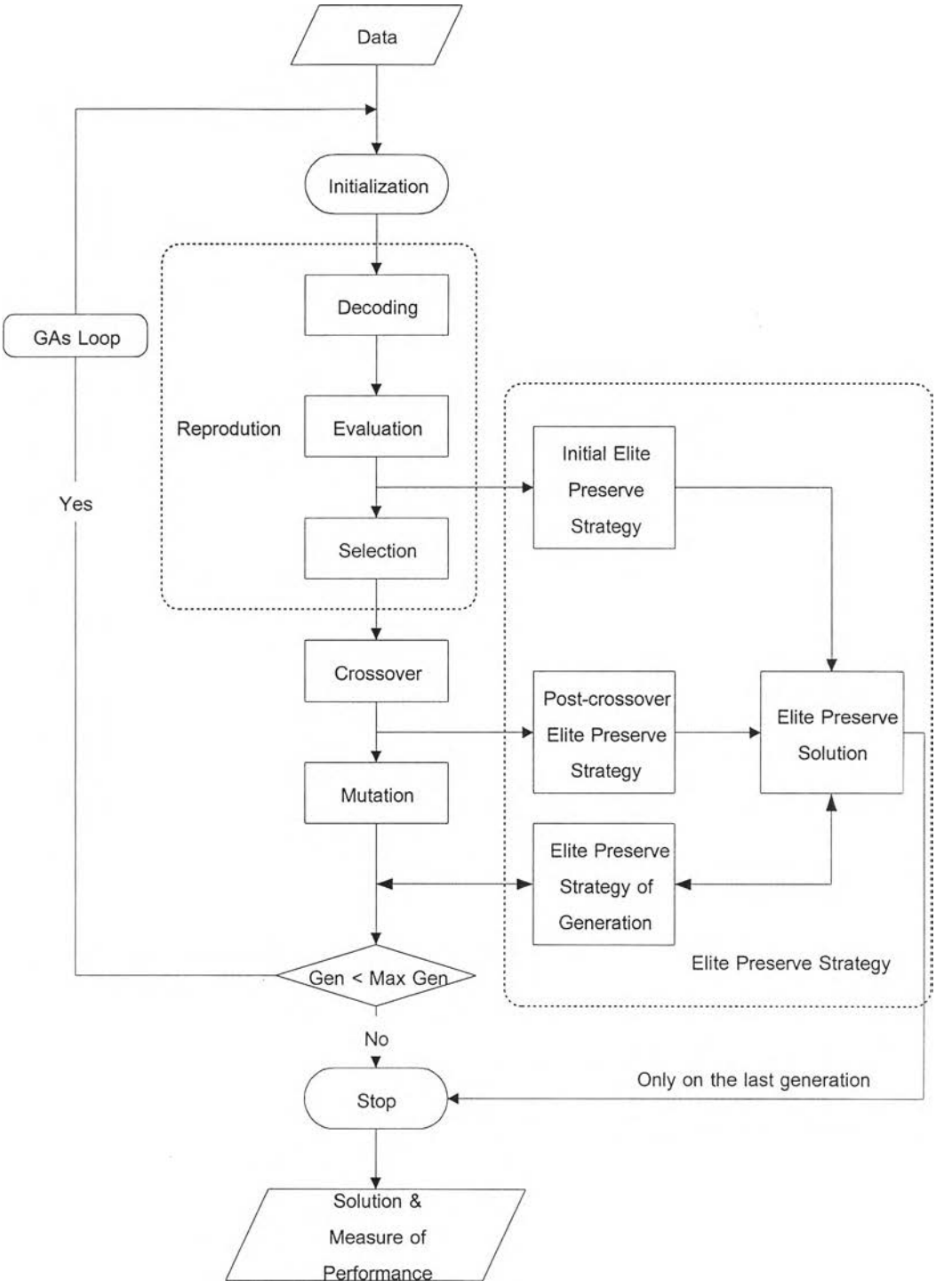
6.3 ขั้นตอนการทำงานของเจนเนติกอัลกอริทึม

จากโครงสร้างหลักของเจนเนติกอัลกอริทึม สามารถแบ่งแยกย่อยเป็นวิธีการของเจนเนติกอัลกอริทึม ได้ดังนี้

1. **Data Input :** รับข้อมูลเข้าต่างๆ ซึ่งได้แก่ ชนิดและจำนวนผลิตภัณฑ์ ขั้นงานการทำงานของแต่ละผลิตภัณฑ์ จำนวนสถานีงานและขั้นงานในสถานีงานต่างๆ และเวลาการทำงานในแต่ละขั้นงาน
2. **Representation & Initialization :** นำข้อมูลต่างๆมาสร้างคำตอบเบื้องต้นแบบสุ่มจำนวน *popsize* ตัว โดยผ่านกระบวนการใส่รหัสคำตอบ (Representation) และการสร้างประชากรเบื้องต้น (Initial Population)
3. **Decoding :** นำรหัสคำตอบของประชากรเบื้องต้นทุกตัว มาถอดรหัสคำตอบเพื่อให้ได้คำตอบที่สมบูรณ์และสามารถนำไปคำนวณหาค่าต่างๆที่ต้องการได้
4. **Evaluation :** คำนวณหาค่าต่างๆที่ต้องการ เช่น เวลาการทำงานในแต่ละสถานีงานของแต่ละผลิตภัณฑ์ เวลาการทำงานเสร็จสิ้น และค่าความพึงพอใจของเวลาการทำงานเสร็จสิ้น ซึ่งจะแสดงออกมาเป็นค่า Fitness Value ของลำดับผลิตภัณฑ์แต่ละแบบที่จัดได้ โดยลำดับผลิตภัณฑ์ที่ทำให้เวลาการทำงานเสร็จสิ้นน้อยที่สุดจะมีค่า Fitness Value ที่สูงสุด
5. **Initial Elite Preserve Strategy :** หาค่าคำตอบที่ดีที่สุดจากประชากรเจนเนอเรชั่นแรกและเก็บค่าคำตอบที่ดีที่สุดนี้ไว้เป็น Elite Preserve Solution

6. **Selection :** คัดเลือกคำตอบที่ดีเข้าสู่ Mating Pool เพื่อเตรียมทำการจับคู่ โดยอาศัยวิธี Selection หาคำตอบที่มีความเหมาะสมมากกว่า
7. **Crossover :** ทำการจับคู่คำตอบที่อยู่ใน Mating Pool และทำการครอสโอเวอร์ด้วยความน่าจะเป็นในการครอสโอเวอร์ เท่ากับ Pc
8. **Post-crossover Elite Preserve Strategy :** หาค่าคำตอบที่ดีที่สุดจากการครอสโอเวอร์ แล้วนำมาเปรียบเทียบกับ Elite Preserve Solution ที่มีอยู่ ถ้าคำตอบที่ได้จากการ ครอสโอเวอร์ดีกว่า ก็ให้เก็บคำตอบนั้นเป็น Elite Preserve Solution แทน
9. **Mutation :** ทำการมิวเตชั่นสตริงคำตอบที่มีด้วยความน่าจะเป็นในการมิวเตชั่น เท่ากับ Pm
10. **Elite Preserve Strategy of Generation :** หาค่าคำตอบที่ดีที่สุดจากการมิวเตชั่น แล้วนำมาเปรียบเทียบกับ Elite Preserve Solution ที่มีอยู่ ถ้าคำตอบที่ได้จากมิวเตชั่นดีกว่า ก็ให้เก็บคำตอบนั้นเป็น Elite Preserve Solution แทน แต่ถ้า Elite Preserve Solution ดีกว่า ก็ให้แทนที่คำตอบที่แย่ที่สุดจากการมิวเตชั่นด้วย Elite Preserve Solution
11. **GAs-loop :** ดูว่า เจนเนอเรชั่น น้อยกว่าจำนวนเจนเนอเรชั่นสูงสุดหรือไม่ ถ้าน้อยกว่าให้กลับไปทำข้อที่ 7-10 ถ้าไม่ให้ทำข้อที่ 12
12. **Stop :** หยุดกระบวนการของเจนเนติกอัลกอริทึม และนำค่า Elite Preserve Solution มาเป็นคำตอบ

โครงสร้างและขั้นตอนของเจนเนติกอัลกอริทึม แสดงได้ดังในรูปที่ 6.1



รูปที่ 6.1 แผนผังแสดงโครงสร้างและขั้นตอนของเจนเนติกอัลกอริทึม

6.4 วิธีการของฟัซซี่เจนเนติกอัลกอริทึมสำหรับปัญหาการจัดลำดับงานเข้าสายการประกอบแบบผลิตภัณฑ์ผสมที่มีเวลาการทำงานแบบฟัซซี่

6.4.1 การใส่รหัสคำตอบ

การใส่รหัสคำตอบ (Chromosome Representation/Coding) คือ การเปลี่ยนคำตอบของปัญหาให้อยู่ในรูปของสตริงคำตอบ หรือที่เรียกว่า Chromosome ซึ่งเป็นขั้นตอนแรกของเจนเนติกอัลกอริทึม ซึ่งถือว่าเป็นขั้นตอนที่สำคัญและมีผลอย่างมากต่อขั้นตอนอื่นๆของ GAs วิธีการใส่รหัสคำตอบมีทั้งแบบ Binary String และ Non-binary String ในกรณีของปัญหาการจัดลำดับผลิตภัณฑ์เข้าสายงานการประกอบแบบผลิตภัณฑ์ผสม คำตอบของปัญหา คือ ลำดับของผลิตภัณฑ์ที่จะเข้าทำการประกอบในสายการประกอบ ดังนั้น วิธีการใส่รหัสคำตอบที่ใช้จึงต้องสามารถแสดงลำดับของผลิตภัณฑ์ในรูปของสตริงได้ วิธีการใส่รหัสคำตอบที่ใช้จึงเป็นแบบ Non-binary String

การจัดลำดับผลิตภัณฑ์เข้าสายการประกอบที่มีลักษณะเป็นสายการประกอบแบบผลิตภัณฑ์ผสม จะจัดผลิตภัณฑ์เข้าเป็นกลุ่ม โดยใช้สัดส่วนของผลิตภัณฑ์ที่ทำการผลิต (Minimum Part Set: MPS) ซึ่งเป็นการจัดลำดับที่ตอบสนองความต้องการด้านความหลากหลายของรูปแบบผลิตภัณฑ์อย่างเหมาะสม โดยลักษณะของสตริงคำตอบ มีดังนี้

1) คำตอบ 1 คำตอบแทนด้วยสตริงคำตอบ 1 ตัว ที่เรียกว่า สตริง
2) ใน 1 สตริง จะแบ่งเป็นหน่วยเล็กๆที่เรียกว่า bit เรียงกันอยู่ โดยจำนวนของ bit จะเท่ากับจำนวนกลุ่มของผลิตภัณฑ์ทั้งหมดที่ต้องเข้าทำการผลิตบนสายการประกอบที่พิจารณา
3) ในแต่ละ bit จะมีค่าตัวเลขตั้งแต่ 1 ถึง n เมื่อ n คือ จำนวนชนิดผลิตภัณฑ์ที่จะทำการผลิต
4) ตำแหน่งของ bit หมายถึงลำดับของผลิตภัณฑ์
5) ตัวเลขในแต่ละ bit จะซ้ำกันได้ โดยจำนวนตัวเลขที่เหมือนกันจะมีจำนวนซ้ำเท่ากับจำนวน MPS ของแต่ละผลิตภัณฑ์

6.1.4.1 การสร้างประชากรเบื้องต้น

การสร้างประชากรเบื้องต้น *(Initial Population Creating)* คือ การสร้างสตริงคำตอบ ซึ่งตามหลักการของ GAs มักใช้วิธีการแบบสุ่ม หมายความว่าตัวเลขที่นำมาใส่ในแต่ละ bit จะต้องถูกเลือกแบบสุ่ม แต่อย่างไรก็ตามในการสร้างสตริงคำตอบสำหรับปัญหาการจัดลำดับผลิตภัณฑ์เข้าสายการประกอบแบบผลิตภัณฑ์ผสม สตริงคำตอบที่ได้ต้องสอดคล้องกับ

ลักษณะปัญหา นั่นคือ สตริงคำตอบแสดงถึงลำดับผลิตภัณฑ์ที่จะเข้าทำการประกอบในสายการประกอบแบบผลิตภัณฑ์ผสม ดังนั้นข้อมูลที่จำเป็นสำหรับการสร้างสตริงคำตอบ ได้แก่ จำนวนชนิดผลิตภัณฑ์ และจำนวนการผลิตของผลิตภัณฑ์แต่ละชนิด

ขั้นตอนการสร้างสตริงคำตอบ มีดังนี้

1. การสร้างตัวเลขในสตริงคำตอบ

ตัวเลขในสตริงคำตอบ ได้แก่ จำนวนชนิดผลิตภัณฑ์ที่จะทำการผลิต เช่น ต้องการผลิตผลิตภัณฑ์ 3 ชนิด ดังนั้นได้ว่า ตัวเลขที่จะประกอบขึ้นเป็นสตริงคำตอบ จะประกอบด้วยเลข 1 2 และ 3 ซึ่งแสดงถึงชนิดของผลิตภัณฑ์

2. จำนวนซ้ำของตัวเลขแต่ละตัวในสตริงคำตอบ

หาได้จากการหาค่าสัดส่วนของผลิตภัณฑ์ที่ทำการผลิต (MPS) เช่น ต้องการผลิตผลิตภัณฑ์ 3 ชนิด โดยผลิตชนิดที่ 1 2 และ 3 เป็นจำนวน 20 50 และ 10 ตามลำดับ ดังนั้นค่าสัดส่วนของผลิตภัณฑ์คือ 1:2:3 = 2:5:1 นั่นคือ ได้ว่า ในสตริงคำตอบ จะประกอบไปด้วยตัวเลข 3 ตัว ซึ่งเป็นตัวแทนชนิดผลิตภัณฑ์ คือ 1 2 และ 3 ซึ่งมีจำนวนซ้ำของตัวเลขเท่ากับ 2 5 และ1 ตามลำดับ

3. การหาความยาวของสตริงคำตอบ

ความยาวของสตริงคำตอบ คำนวณได้จากค่าผลรวมของ MPS ที่หาได้ในขั้นตอนที่ 2 คือ 2+5+1 = 8 นั่นคือ สตริงคำตอบมีความยาวเท่ากับ 8 bit

จากทั้ง 3 ขั้นตอน ทำให้ได้ว่าสตริงคำตอบตัวอย่าง มีความยาว 8 bit ประกอบด้วยตัวเลข 1 2 และ 3 เป็นจำนวน 2 5 และ 1 ตัว ตามลำดับ โดยตัวเลขแสดงถึงชนิดผลิตภัณฑ์ และตำแหน่งของ bit คือลำดับของผลิตภัณฑ์ที่เข้าทำการผลิต ซึ่งตัวเลขที่แสดงในแต่ละตำแหน่ง จะได้จากการสุ่ม ตัวอย่างสตริงคำตอบที่เป็นไปได้ คือ 1-2-2-1-2-3-2-2 นั่นคือ ทำการผลิตผลิตภัณฑ์ชนิดที่ 1 2 1 2 3 และ 2 ตามลำดับ สลับชนิดไปมา เนื่องจากเป็นสายการประกอบแบบผลิตภัณฑ์ผสม

6.4.1.2 จำนวนประชากรเบื้องต้น

จำนวนประชากรเบื้องต้น *(Population Size)* คือ จำนวนคำตอบเบื้องต้นที่สร้างขึ้นขึ้นมาจำนวนหนึ่งเพื่อนำไปใช้ในกระบวนการของ GAs โดยคำตอบ 1 คำตอบคือประชากร 1 ตัว จำนวนประชากรเบื้องต้นจะเท่ากับจำนวนประชากรในแต่ละเจนเนอเรชั่น และเป็นพารามิเตอร์ตัวหนึ่งที่มีผลต่อประสิทธิภาพของ GAs ดังนั้นต้องมีการกำหนดจำนวนประชากรเบื้องต้นที่จะสร้างขึ้นในแต่ละเจนเนอเรชั่น ซึ่งในที่นี้กำหนดให้เท่ากับ *popsize* ตัว

การสร้างประชากรเบื้องต้น จะสร้างตามขั้นตอนการสร้างสตริงคำตอบดังที่กล่าวข้างต้น โดยสร้างประชากรเบื้องต้นขึ้นทั้ง *popsize* ตัว ต้องไม่ซ้ำกัน เพื่อให้เกิดความหลากหลายของคำตอบ และเป็นการป้องกันไม่ให้คำตอบที่ได้จากวิธีการของเจนเนติกอัลกอริทึมที่ใช้เป็นค่า Local Optimal นอกจากนี้การสร้างประชากรเบื้องต้นให้แตกต่างกันยังช่วยให้สามารถกำหนดจำนวนประชากรน้อยลงได้

6.4.2 การถอดรหัสคำตอบ

การถอดรหัสคำตอบ (Decoding) เป็นการแปลงค่าที่ได้จากสตริงคำตอบ โดยในงานวิจัยนี้สตริงคำตอบที่ได้ แสดงถึงลำดับผลิตภัณฑ์ที่เข้าทำการประกอบในสายการประกอบแบบผลิตภัณฑ์ผสม ดังนั้นในการถอดรหัสคำตอบจากสตริงคำตอบ ทำให้ได้ลำดับผลิตภัณฑ์ที่จะทำการผลิต เช่น จากปัญหาการจัดลำดับผลิตภัณฑ์ ชนิดที่ 1 2 และ 3 ที่ต้องการทำการผลิตจำนวน 20 50 และ 10 ชิ้น ตามลำดับ ได้ว่าสัดส่วนของการผลิตผลิตภัณฑ์ เท่ากับ 2:5:1 ดังนั้นในการจัดลำดับผลิตภัณฑ์เข้าสายการประกอบจะจัดเข้าทำการผลิตเป็นกลุ่ม กลุ่มละ 10 ชิ้น หากสตริงคำตอบที่ได้เป็น 1-2-2-1-2-3-2-2 แสดงถึงลำดับผลิตภัณฑ์ที่จะเข้าทำการประกอบโดยทำการผลิตผลิตภัณฑ์ชนิดที่ 1 2 2 1 2 3 2 และ 2 ตามลำดับ ครั้งละ 10 ชิ้น แสดงให้เห็นว่าในสายการประกอบเดียวกัน จะมีการผลิตผลิตภัณฑ์ชนิดต่างกัน โดยมีการสลับชนิดของผลิตภัณฑ์ที่เข้าทำการผลิต ซึ่งสตริงคำตอบที่ต่างกัน จะแสดงถึงความแตกต่างของลำดับผลิตภัณฑ์ที่จะเข้าในสายการประกอบที่ต่างกัน เช่น1-2-2-1-2-3-2-2 และ2-2-1-2-3-1-2-2

6.4.3 การประเมินค่า

การประเมินค่า (Evaluation) เป็นขั้นตอนการคำนวณเพื่อประเมินความเหมาะสมของสตริงคำตอบแต่ละตัวว่ามีความเหมาะสมมากน้อยเพียงใด โดยนำหลักการทางฟัซซี่มาใช้ในการหาค่า Fitness แสดงถึงความพึงพอใจของผู้ทำการตัดสินใจในการจัดลำดับผลิตภัณฑ์ในเรื่องเวลาการทำงานเสร็จสิ้นซึ่งเป็นค่าของฟังก์ชั่นวัตถุประสงค์ โดยหากสตริงคำตอบใดมีค่า Fitness มาก ก็หมายความว่ามีความเหมาะสมมาก

ขั้นตอนการประเมินค่า

1. รวมเวลาการทำงานของทุกๆผลิตภัณฑ์ ในทุกๆสถานีงาน โดยใช้หลักการรวมแบบฟัซซี่ เนื่องจากในสถานีงานต่างๆ ประกอบด้วยหลายๆ ขั้นงาน ข้อมูลเข้า (Input Data) ที่รับเข้ามาเป็นเวลาการทำงานของแต่ละขั้นงาน ดังนั้นจึงหาเวลาการทำงานของแต่

ละผลิตภัณฑ์ในแต่ละสถานีงานโดยการรวมเวลาการทำงานของขั้นงานต่างๆ ในแต่ละสถานีงานดังแสดงในสมการที่ 6.1

$$(A_{nj}, B_{nj}, C_{nj}) = (a_{n1}+a_{n2}+\ldots a_{ni}, b_{n1}+b_{n2}+\ldots b_{ni}, c_{n1}+c_{n2}+\ldots c_{ni}) \quad (6.1)$$

เมื่อ (A_{nj}, B_{nj}, C_{nj}) คือ เวลาการประกอบ (น้อยที่สุด เฉลี่ย มากที่สุด) ของผลิตภัณฑ์ n บนสถานีงาน j

(a_{ni}, b_{ni}, c_{ni}) คือ เวลาการประกอบ(น้อยที่สุด เฉลี่ย มากที่สุด) ของผลิตภัณฑ์ n ในขั้นงาน i

2. จัดลำดับผลิตภัณฑ์เข้าสายการประกอบตามลำดับผลิตภัณฑ์ที่ได้จากสตริงคำตอบ โดยพิจารณาเหมือนการจัดลำดับงานของสายการผลิตแบบ Flow Shop ซึ่งจะทำให้เกิดเวลาว่างในแต่ละสถานีงาน

3. หาค่าเวลาการประกอบทุกผลิตภัณฑ์เสร็จสิ้นในแต่ละสถานีงาน โดยการรวมเวลาการทำงานและเวลาว่าง ซึ่งมีลักษณะเป็นฟัซซี่เซต ดังแสดงในสมการที่ 6.2

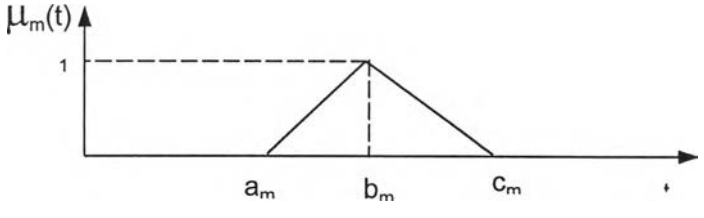
$$\mu_m(t) = \sum^{n}_{k=1}\mu_{[k]m}(t) + \sum^{n}_{k=1}\mu_{idle[k]m}(t) \quad (6.2)$$

เมื่อ $\mu_m(t)$ = ฟังก์ชันการเป็นสมาชิกของเวลาการทำงานเสร็จสิ้น

$\mu_{[j]m}(t)$ = ฟังก์ชันการเป็นสมาชิกของเวลาการทำงานของผลิตภัณฑ์ k บนเครื่องจักรตัวที่ m

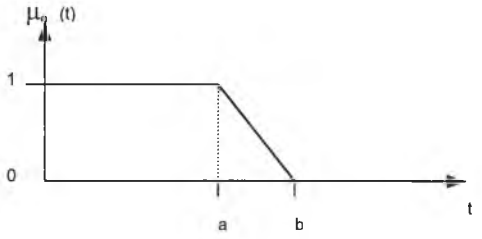
$\mu_{idle[k]m}(t)$ = ฟังก์ชันการเป็นสมาชิกของเวลาว่างของผลิตภัณฑ์ k บนเครื่องจักรตัวที่ m

4. พิจารณาเวลาจากสถานีงานที่มีเวลาการประกอบทุกผลิตภัณฑ์เสร็จสิ้นมากที่สุด เป็นเวลาที่ใช้ในการประกอบทั้งหมด โดยจะอยู่ในรูปฟัซซี่เซตมีลักษณะฟังก์ชันการเป็นสมาชิกเป็นรูปสามเหลี่ยม ดังแสดงในรูปที่ 6.2



***รูปที่ 6.2** กราฟแสดงฟังก์ชันการเป็นสมาชิกของเวลาการประกอบเสร็จสิ้น*

5. สร้างฟังก์ชันการเป็นสมาชิกของวัตถุประสงค์ คือ ฟังก์ชันของค่าความพึงพอใจ ได้ตามสมการที่ 6.3 ดังแสดงในรูปที่ 6.3



รูปที่ 6.3 กราฟแสดงฟังก์ชันการเป็นสมาชิกของวัตถุประสงค์

$$\mu_o(t) = \begin{cases} 1 & \text{เมื่อ } t \le a \\ \dfrac{b-t}{b-a} & \text{เมื่อ } a < t < b \\ 0 & \text{เมื่อ } t \ge b \end{cases} \qquad (6.3)$$

เมื่อ $\mu_o(t)$ = ค่าความพอใจของเวลาในการทำงานเสร็จสิ้น

a = เวลาการทำงานเสร็จสิ้นที่ผู้ตัดสินใจพึงพอใจสูงสุด
(ความพอใจ = 1 และลดลงเมื่อใช้เวลามากกว่านี้)

b = เวลาการทำงานเสร็จสิ้นที่มากที่สุดที่ผู้ตัดสินใจยอมรับได้
(หากเกินกว่านี้จะไม่ยอมรับคือค่าความพอใจ = 0)

หมายเหตุ : ค่าของ a กำหนดจากผลบวกของค่าเฉลี่ยของเวลาทำงานทุกขั้นงาน
ค่าของ b กำหนดจากผลบวกของค่ามากที่สุดของเวลาการทำงานทุกขั้นงาน

6. หาค่า Fitness ซึ่งเป็นค่าแสดงความพึงพอใจของผู้จัดลำดับผลิตภัณฑ์ต่อเวลาการทำงานเสร็จสิ้น โดยนำเอาฟังก์ชันเวลาการทำงานเสร็จสิ้นและฟังก์ชันของวัตถุประสงค์มาผ่านกระบวนการ Max-Min Operator ดังแสดงในสมการที่ 6.4 ทำให้ได้ค่า Fitness ของสตริงคำตอบแต่ละตัว

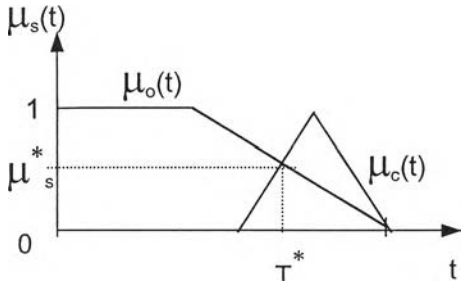
$$max\ \mu_s(t) = max\{min\{\mu_0(t),\ \mu_c(t)\}\} \qquad (6.4)$$

เมื่อ μ_s = ค่าความพอใจของเวลาจากการผลิตตามลำดับผลิตภัณฑ์ที่จัดไว้

$\mu_0(t)$ = ฟังก์ชันการเป็นสมาชิกของวัตถุประสงค์

$\mu_c(t)$ = ฟังก์ชันการเป็นสมาชิกของเวลาทำงานเสร็จสิ้น

การหา Fitness โดยวิธี Max-Min Operator แสดงได้ดังในรูปที่ 6.4



รูปที่ 6.4 กราฟแสดงการหาค่า Fitness โดยใช้ Max-Min Operator

เมื่อ μ^*_s = ค่าความพอใจของเวลาจากการผลิตตามลำดับผลิตภัณฑ์ที่จัดไว้ (ซึ่งเป็นค่า Fitness ของสตริงคำตอบแต่ละตัว)

T^* = เวลาการทำงานที่ทำให้ได้ค่าความพึงพอใจเท่ากับ μ^*_s

7. หาค่า Fitness ของสตริงคำตอบแต่ละตัว ตามขั้นตอนที่ 2-6 ทำให้ได้ค่า Fitness ที่แสดงถึงค่าความเหมาะสมของสตริงคำตอบแต่ละตัว สตริงคำตอบที่มีค่า Fitness มากที่สุดเป็นสตริงคำตอบที่มีความเหมาะสมมากที่สุด นำค่า Fitness ที่ได้ไปใช้ในกระบวนการคัดเลือกสตริงซึ่งเป็นขั้นตอนต่อไปของเจนเนติกอัลกอริทึม

6.4.4 การคัดเลือกคำตอบ

การคัดเลือกคำตอบ (Selection) เป็นการคัดเลือกสตริงคำตอบที่มีความเหมาะสมเพื่อเข้าไปสู่ขั้นตอนต่อไปของ GAs โดยพิจารณาจากค่า Fitness ถ้าสตริงคำตอบใดมีค่า Fitness มาก ก็มีโอกาสที่จะถูกคัดเลือกไว้มากกว่าตัวที่มีค่า Fitness น้อย โดยจะคัดเลือกสตริงคำตอบจำนวน *popsize* ตัว

วิธีการคัดเลือกคำตอบ ที่นำมาพิจารณาใช้ในงานวิจัยนี้ มี 2 วิธี คือ วิธี Roulette Wheel Selection (Holland,1975) และ วิธี Tournament Selection (Goldberg,1991)

6.4.4.1 วิธี Roulette Wheel Selection

วิธีการคัดเลือกโดยใช้วิธี Roulette Wheel Selection เป็นวิธีการคัดเลือกสตริงที่นับว่าง่ายวิธีหนึ่ง โดยมีการสร้างวงล้อรูเล็ต แล้วทำการสุ่มตัวเลข หากตัวเลขที่สุ่มขึ้นมาตรงกับสัดส่วนใดของวงล้อรูเล็ต สตริงคำตอบที่อยู่ภายใต้พื้นที่ของวงล้อรูเล็ตนั้นจะได้รับการคัดเลือกให้เข้าสู่ Mating Pool เพื่อผ่านกระบวนการต่อไปของ Gas

วิธีการสร้างวงล้อรูเล็ต

วงล้อรูเล็ต (Roulette Wheel) คือวงกลมที่มีพื้นที่ขนาด 1 หน่วย ซึ่งพื้นที่ถูกแบ่งออกเป็นส่วนๆ ตามจำนวนของประชากรในแต่ละเจนเนอเรชั่น (เท่ากับ *popsize* ส่วน) พื้นที่แต่ละส่วนจะมีขนาดเท่ากับความน่าจะเป็นในการถูกเลือกของสตริงคำตอบแต่ละตัว ซึ่งมีวิธีการสร้างและวิธีการเลือกสตริงคำตอบดังนี้

1. หาค่า Fitness รวมของสตริงคำตอบทั้งหมด *popsize* ตัว ดังสมการที่ 6.5

$$F = \sum_{i=1}^{popsize} f(Xi) \qquad (6.5)$$

โดย $f(Xi)$ = ค่า Fitness ของสตริงตัวที่ i

2. หาค่าความน่าจะเป็นในการถูกคัดเลือก (Probability of Selection) ของสตริงคำตอบแต่ละตัว ตามสมการที่ 6.6

$$p_i = \frac{f(Xi)}{F} \qquad i = 1, 2, \ldots, popsize \qquad (6.6)$$

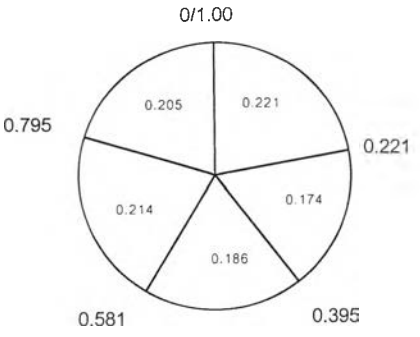
3. หาค่าความน่าจะเป็นในการถูกคัดเลือกสะสม (Cumulative Probability of Selection) ของสตริงคำตอบแต่ละตัว ตามสมการที่ 6.7

$$q_i = \sum_{j=1}^{i} p_j \qquad (6.7)$$

ตัวอย่างของวงล้อรูเล็ตแสดงได้ดังตารางที่ 6.1 และรูปที่ 6.5

ตารางที่ 6.1 *ตัวอย่างตารางแสดงการสร้างวงล้อรูเล็ต*

String No.	Fitness	*pi*	*qi*
1	0.950	0.221	0.221
2	0.750	0.174	0.395
3	0.800	0.186	0.581
4	0.920	0.214	0.795
5	0.885	0.205	1.000
รวม	4.305	1.000	



รูปที่ 6.5 วงล้อรูเล็ต

4. ทำการสร้างตัวเลขสุ่ม r ซึ่งมีค่าระหว่าง 0 ถึง 1 ขึ้นมา 1 ค่า คือ r_1
   - ถ้า $r_1 < q_1$ ให้เลือกสตริงคำตอบตัวที่ 1
   - ถ้า $q_{i-1} < r_1 < q_i$ (เมื่อ $2 < i < popsize$) ให้เลือกสตริงคำตอบตัวที่ i เข้าสู่ Mating Pool

   ตัวอย่างเช่น หากตัวเลขสุ่ม r_1 มีค่าเท่ากับ 0.185 ก็นำมาเทียบกับค่า q_i ปรากฏว่ามีค่าไม่เกินค่า q_1 (0.221) ก็จะเลือกสตริงคำตอบตัวที่ 1 (String No.1) เข้าสู่ Mating Pool แต่หากตัวเลขสุ่ม r_1 มีค่าเท่ากับ 0.650 แล้วเมื่อนำมาเทียบกับค่า q_i จะพบว่าอยู่ในค่าระหว่าง q_3 และ q_4 ($q_3 < r_1 < q_4$) ดังนั้นจึงเลือกสตริงคำตอบตัวที่ 4 (String No. 4) เข้าสู่ Mating Pool
5. ทำตามขั้นตอนข้อที่ 4 จนกว่าจะได้สตริงคำตอบใน Mating Pool ครบ *popsize* ตัว

- ข้อดีของวิธี Roulette Wheel Selection คือ การได้สตริงคำตอบที่หลากหลายเพื่อนำไปใช้ในการพัฒนาคำตอบ เนื่องจากสตริงคำตอบที่ได้รับคัดเลือกจะขึ้นกับตัวเลขที่ทำการสุ่มได้

- ข้อเสียของวิธี Roulette Wheel Selection คือ สตริงคำตอบที่มีค่า Fitness น้อยจะถูกคัดเข้าไปสู่ Mating Pool ซึ่งอาจจะทำให้คำตอบที่ได้ทำการพัฒนาจากค่าของสตริงเริ่มต้น มีโอกาสที่จะได้ค่าที่ดีขึ้นหรือแย่ลงก็ได้

6.4.4.2 วิธี Tournament Selection

เป็นวิธีการที่ดัดแปลงมาจากวิธี Roulette Wheel Selection โดยสุ่มสตริงคำตอบจากวงล้อรูเล็ตมา 2 ตัว แล้วนำค่า Fitness มาเปรียบเทียบกันอีกครั้งหนึ่ง ทำให้ได้สตริงคำตอบที่มีความเหมาะสมมากกว่าวิธีการ Roulette Wheel Selection ซึ่ง มีโอกาสสุ่มสตริงคำตอบที่มีค่า Fitness น้อยๆด้วย

ขั้นตอนการคัดเลือกสตริงคำตอบโดยวิธี Tournament Selection มีดังนี้

1. สร้างตัวเลขสุ่ม r ซึ่งมีค่าระหว่าง 0 ถึง 1 ขึ้นมา 1 ค่า คือ r_1
   - ถ้า $r_1 < q_1$ ให้เลือกสตริงคำตอบตัวแรก
   - ถ้า $q_{i-1} < r_1 < q_i$ (เมื่อ $2< i < popsize$) ให้เลือกสตริงคำตอบตัวที่ i มาเป็นสตริงคำตอบตัวแรก
2. สร้างตัวเลขสุ่ม r ซึ่งมีค่าระหว่าง 0 ถึง 1 ขึ้นมาอีก 1 ค่า คือ r_2
   - ถ้า $r_2 < q_1$ ให้เลือกสตริงคำตอบตัวแรก
   - ถ้า $q_{i-1} < r2 < q_i$ (เมื่อ $2< i < popsize$) ให้เลือกสตริงคำตอบตัวที่ i มาเป็นสตริงคำตอบตัวที่สอง
3. นำค่า Fitness ของสตริงคำตอบทั้ง 2 ตัวมาเปรียบเทียบกัน ตัวใดมีค่า Fitness มากกว่าก็ให้เลือกตัวนั้นเข้าสู่ Mating Pool ตัวอย่างเช่น สตริงคำตอบตัวแรกที่ได้คือ String No. 1 และสตริงคำตอบตัวที่สองที่ได้คือ String No. 2 เมื่อนำค่า Fitness ของสตริงทั้ง 2 ตัวมาเปรียบเทียบกันจะพบว่า String No. 2 จะมีค่า Fitness มากกว่า จึงเลือก String No. 2 เข้าสู่ Mating Pool
4. ทำตามขั้นตอนข้อที่ 1-3 จนกว่าจะได้สตริงคำตอบใน Mating Pool ครบ *popsize* ตัว

จากวิธีดังกล่าวจะเห็นได้ว่า สตริงคำตอบที่มีค่า Fitness มากก็จะมีพื้นที่มาก จึงมีโอกาสที่ตัวเลขสุ่มที่สร้างขึ้นมาจะตกอยู่ภายในบริเวณของสตริงคำตอบตัวนั้นมากกว่าตัวที่มีค่า Fitnessน้อย (มีพื้นที่น้อย) ทำให้สตริงคำตอบที่ถูกเลือกเข้าสู่ Mating Pool เป็นสตริงคำตอบที่มีค่า Fitness โดยเฉลี่ยสูงกว่าสตริงคำตอบเดิม

ตารางที่ 6.2 *ตัวอย่างการคัดเลือกด้วยวิธี Tournament Selection*

ครั้งที่	ประชากรตัวที่ 1				ประชากรตัวที่ 2				หมายเลขประชากรที่เลือก
	r_1	$q_1 > r_1$	หมายเลขประชากร	ค่า Fitness	r_2	$q_1 > r_2$	หมายเลขประชากร	ค่า Fitness	
1	0.320	0.395	2	0.750	0.951	1.000	5	0.885	5
2	0.178	0.221	1	0.950	0.607	0.653	4	0.920	1
3	0.891	1.000	5	0.885	0.762	0.159	1	0.950	1
4	0.457	0.581	3	0.800	0.018	0.200	1	0.950	1
5	0.658	0.795	4	0.920	0.406	0.493	3	0.750	4

(หมายเหตุ: qi ที่มากกว่าค่า r ในคอลัมน์ที่ 3 และ7 ได้มาจากตารางที่ 6.1)

จากตารางที่ 6.2 แสดงตัวอย่างการคัดเลือกด้วยวิธี Tournament Selection ซึ่งจะเห็นได้ว่าสตริงคำตอบหมายเลข 1 ซึ่งมีค่า Fitness มากที่สุด จะถูกสุ่มเลือกขึ้นมามากครั้งที่สุด ใน

ขณะที่สตริงคำตอบซึ่งมีค่า Fitness น้อยก็จะถูกสุ่มเลือกน้อยครั้งเช่นกัน ข้อสังเกตประการหนึ่งจากตัวอย่างก็คือ ในการสุ่ม ก็สุ่มได้สตริงคำตอบหมายเลข 2 ที่มีค่า Fitness น้อยที่สุดด้วย ซึ่งถ้าใช้วิธี Roulette Wheel สตริงคำตอบหมายเลข 2 นี้ก็จะมีโอกาสที่จะถูกเลือกเข้าสู่ Mating Pool และจะได้รับการดำเนินการตามกระบวนการ GAs ต่อไป แม้ว่าสตริงคำตอบตัวนี้จะมีความเหมาะสมต่ำก็ตาม แต่เมื่อใช้วิธีคัดเลือกแบบ Tournament Selection สตริงคำตอบหมายเลข 2 นี้จะต้องถูกนำไปเปรียบเทียบกับสตริงคำตอบอีกตัวก่อน ดังนั้นโอกาสที่สตริงคำตอบตัวนี้จะถูกเลือกเข้าสู่ Mating Pool ก็จะลดลง

- ข้อดีของวิธี Tournament Selection คือ สตริงที่ถูกคัดเลือกจะเป็นสตริงคำตอบที่ดีกว่า ทำให้โอกาสในการพัฒนาคำตอบที่ดีขึ้นเป็นไปได้สูงเสมือนเป็นการพัฒนาคำตอบไปในตัว
- ข้อเสียของวิธี Tournament Selection คือ คำตอบที่ได้อาจจะติดอยู่ใน Local Optimal ซึ่งเป็นคำตอบที่ดีแต่ไม่ใช่คำตอบที่ดีที่สุด เนื่องจากโอกาสที่จะคัดสตริงตัวเดียวกันเข้าไปสู่ Mating Pool จะมีสูงทำให้การ Crossover มีประสิทธิภาพที่ลดลง หรืออาจจะไม่มีผลหากมีสตริงตัวเดียวกันอยู่ใน Mating Pool ทั้งหมด

วิธีการคัดเลือกสตริงทั้ง 2 วิธีดังกล่าว มีทั้งข้อดีและข้อเสีย ดังนั้นในการเลือกวิธีการคัดเลือกคำตอบ เพื่อนำไปประยุกต์ใช้อย่างเหมาะสมกับปัญหาที่แตกต่างกันควรมีการทดสอบเพื่อเปรียบเทียบ ทั้ง 2 สองวิธี

6.4.5 การครอสโอเวอร์

การครอสโอเวอร์ (Crossover) เป็นขั้นตอนที่กระทำภายหลังจากการรีโปรดักชั่น ซึ่งถือได้ว่าเป็นส่วนหนึ่งในวิธีการพัฒนาคำตอบของเจนเนติกอัลกอริทึม

การครอสโอเวอร์มีปัจจัยที่เกี่ยวข้องคือ

- วิธีการครอสโอเวอร์ (Crossover Type) มีด้วยกันหลายวิธี โดยในงานวิจัยนี้ ได้นำเสนอวิธีการครอสโอเวอร์ ไว้ทั้งหมด 4 วิธี คือ 1.Modified One point Crossover (modMOX) 2.Modified Partially Mapped Crossover (modPMX) 3.Modified Order Crossover (modOX) และ 4.Modified Position Base Crossover (modPBX)
- ความน่าจะเป็นในการครอสโอเวอร์ (Crossover Probability : Pc) คือ ค่าที่ใช้ในการคัดเลือกสตริงที่จะเข้าสู่การครอสโอเวอร์ โดยเปรียบเทียบกับค่าที่สุ่มขึ้นกับค่า Pc ถ้าสตริงตัวใดที่มีค่าสุ่มน้อยกว่าค่า Pc จะถูกนำไปจับเป็นคู่ กลายเป็นสตริงพ่อแม่ (Parent)
- จำนวนสตริงคำตอบที่ถูกนำมาจับคู่ (Nc) คือจำนวนสตริงที่จะเข้าสู่การครอสโอเวอร์

6.4.5.1 การจับคู่สตริงคำตอบ

จากสตริงคำตอบจำนวน *popsize* ตัวที่ได้มาจากกระบวนการคัดเลือก จะมีสตริงคำตอบเพียงบางส่วนเท่านั้นที่จะถูกนำมาจับคู่เพื่อเตรียมสำหรับกระบวนการครอสโอเวอร์ สตริงคำตอบที่ไม่ได้ถูกนำไปจับคู่ก็จะยังคงสภาพเดิมและอยู่ใน Mating Pool (เป็นประชากรในเจนเนอเรชั่น) ต่อไป จำนวนสตริงคำตอบที่จะถูกนำมาจับคู่ (Nc) ขึ้นอยู่กับความน่าจะเป็นในการครอสโอเวอร์ (Pc) การจับคู่สตริงคำตอบเพื่อที่จะนำไป ครอสโอเวอร์ มีขั้นตอนดังนี้

1. สร้างตัวเลขสุ่ม r ซึ่งมีค่าระหว่าง 0 ถึง 1 ให้กับสตริงคำตอบแต่ละตัว
2. สตริงคำตอบตัวใดที่ตัวเลขสุ่มมีค่าน้อยกว่า Pc จะถูกเลือกไปจับคู่และทำการครอสโอเวอร์
3. ถ้าไม่มีสตริงคำตอบตัวใดที่มีค่า r น้อยกว่า Pc ให้เริ่มทำข้อ 1 และ 2 อีกครั้ง
4. ถ้ามีสตริงคำตอบที่มีค่า r น้อยกว่า Pc ทั้งหมดจำนวน Nc ตัว โดยที่ Nc เป็นจำนวนคี่ ต้องทำการปรับให้เป็นจำนวนคู่ก่อน โดยมีเงื่อนไขในการปรับ ดังนี้

   - ถ้า Nc เป็นจำนวนคี่ซึ่งมีค่าระหว่าง 1 ถึง *popsize* ให้ทำการสุ่มตัวเลข 0 หรือ 1 ขึ้นมา 1 ค่า ถ้าสุ่มได้เลข 1 ให้เพิ่มสตริงคำตอบเข้าไปอีก 1 ตัว โดยสุ่มเลือกจากตัวที่เหลืออยู่ใน Mating Pool แต่ถ้าสุ่มได้เลข 0 ให้ตัดสตริงคำตอบทิ้ง 1 ตัว โดยสุ่มเลือกจากตัวที่ได้เลือกเอาไว้
   - ถ้า Nc มีค่าเท่ากับ 1 การปรับให้ใช้วิธีเพิ่มสตริงเข้าไปอีก 1 ตัวเท่านั้น
   - ถ้า Nc มีจำนวนเท่ากับ *popsize* ซึ่งเป็นจำนวนคี่ การปรับให้ใช้วิธีตัดสตริงคำตอบที่เตรียมได้ลง 1 ตัวเท่านั้น

5. เมื่อได้สตริงคำตอบที่จะนำมาจับคู่ทั้งหมด Nc ตัว ให้นำมาจับคู่ตามลำดับ ซึ่งจะได้ทั้งหมด $Nc/2$ คู่

6.4.5.2 การครอสโอเวอร์

กระบวนการครอสโอเวอร์ เป็นกระบวนการที่นำสตริงคำตอบที่ถูกจับคู่ไว้ $Nc/2$ คู่ จากขั้นตอนการจับคู่สตริงคำตอบ มาแลกเปลี่ยนสตริงบางส่วนซึ่งกันและกันเพื่อให้เกิดสตริงใหม่ขึ้นโดยจะเรียกสตริงคำตอบ 2 ตัวที่ถูกจับคู่นี้ว่า *"สตริงคำตอบรุ่นพ่อแม่ (Parent)"* และจะเรียกสตริงคำตอบ 2 ตัวที่ได้จากการครอสโอเวอร์นี้ว่า *"สตริงคำตอบรุ่นลูก (Offspring)"* วิธีการครอสโอเวอร์มีหลายวิธี แต่เนื่องจากงานวิจัยนี้ ลักษณะสตริงคำตอบจะมีลักษณะเป็นแบบ Non-binary String ที่มีตัวเลขซ้ำกัน ดังนั้นจึงต้องใช้วิธีการครอสโอเวอร์ที่มี

การดัดแปลงวิธีการที่ใช้อยู่โดยทั่วไป ให้สอดคล้องกับลักษณะของสตริงคำตอบ โดยได้เสนอวิธีการครอสโอเวอร์ไว้ทั้งหมด 4 วิธีดังนี้

1) วิธี Modified One-point Crossover (modMOX)

เป็นวิธีการที่ใช้หลักการเดียวกับวิธี MOX แต่มีการดัดแปลงเพื่อให้สามารถใช้กับสตริงที่มีตัวเลขซ้ำกันได้ เริ่มจากกำหนดจุดครอสโอเวอร์ (Crossover Point : Xp) ขึ้นมา 1 ตำแหน่งโดยการสุ่มค่า ระหว่าง [1,n-1] เมื่อ n คือความยาวของสตริง และกำหนดให้ตำแหน่ง bit ที่ 1 จนถึง Xp ของสตริงพ่อแม่ตัวที่ 1 เป็น H1 ตำแหน่ง bit ตัวที่ 1 ถึง Xp ของ สตริงพ่อแม่ตัวที่ 2 เป็น H2

สมมติสุ่มค่าตัวเลขได้จุดครอสโอเวอร์ เท่ากับ 5 ดังนั้นได้ว่าจุดครอสโอเวอร์ของสตริงพ่อแม่ทั้ง 2 เป็นดังนี้

$$P1 = [\underbrace{2\ 2\ 2\ 1\ 1}_{H1} \mid 1\ 1\ 3\ 1\ 1\ 3\ ]$$

$$P2 = [\underbrace{2\ 3\ 1\ 1\ 1}_{H2} \mid 2\ 3\ 1\ 2\ 1\ 1\ ]$$

นำ H1 มาเป็นส่วนหัวของ สตริงลูกตัวที่ 1 และนำ H2 มาเป็นส่วนหัวของสตริงลูกตัวที่ 2 ตำแหน่งที่เหลือเป็นตำแหน่งว่างแทนด้วย *

$$O1 = [\ 2\ 2\ 2\ 1\ 1 \mid *\ *\ *\ *\ *\ *\ ]$$

$$O2 = [\ 2\ 3\ 1\ 1\ 1 \mid *\ *\ *\ *\ *\ *\ ]$$

ลบค่าใน P2 ที่ซ้ำกับใน O1 ออก โดยเริ่มจากจากตำแหน่งที่ 1 จนครบจำนวนที่ซ้ำกับ O1 โดยการแทน x ลงในตำแหน่งที่ซ้ำใน P2 และลบค่าใน P1 ที่ซ้ำกับใน O2 ออก โดยเริ่มจากจากตำแหน่งที่ 1 จนครบจำนวนที่ซ้ำกับ O2 โดยการแทน x ลงในตำแหน่งที่ซ้ำใน P1

$$O1 = [\ 2\ 2\ 2\ 1\ 1 \mid *\ *\ *\ *\ *\ *\ ]$$

$$P2 = [\ x\ 3\ x\ x\ 1 \mid x\ 3\ 1\ x\ 1\ 1\ ]$$

$$O2 = [\ 2\ 3\ 1\ 1\ 1 \mid *\ *\ *\ *\ *\ *\ ]$$

$$P1 = [\ x\ 2\ 2\ x\ x \mid x\ 1\ x\ 1\ 1\ 3\ ]$$

นำค่าที่เหลืออยู่ใน P2 มาใส่ใน O1 และนำค่าที่เหลืออยู่ใน P1 มาใส่ใน O2 ในตำแหน่งที่ว่างตามลำดับ ดังนั้นได้ว่าสตริงลูกที่ได้จาก คือ

O1 = [2 2 2 1 1 3 1 3 1 1 1]

O2 = [2 3 1 1 1 2 2 1 1 1 3]

2) วิธี Modified Partially Mapped Crossover (modPMX)

เป็นวิธีการครอสโอเวอร์ที่ดัดแปลงมาจาก Partially Mapped Crossover(Goldberg and Lingle,1985) เริ่มต้นโดยการสุ่มตัวเลขมา 2 ตัว ที่เป็นจำนวนเต็มมีค่าอยู่ในช่วง [1,n] เมื่อ n คือความยาวสตริง เพื่อหาสตริงย่อย โดยตัวเลขสุ่มที่มีค่าน้อยกว่าจะเป็นตำแหน่งเริ่มต้นของสตริงย่อย ส่วนตัวเลขสุ่มที่มีค่ามากจะเป็นตำแหน่งสุดท้ายภายในสตริงย่อย ดังนั้นจะได้สตริงย่อยของสตริงพ่อแม่ทั้ง 2 ตัว

สมมติทำการสุ่มได้มา 2 ตัวคือ 4 และ 6 ดังนั้น สตริงย่อยของ P1 = [1 2 3] และสตริงย่อยของ P2 = [3 3 1]

P1 = [1 2 1 | **1 2 3** | 3 1 2 1 1]

P2 = [1 2 2 | **3 3 1** | 1 1 2 1 1]

สตริงลูกได้จากการสลับสตริงย่อยของสตริงพ่อแม่ทั้ง 2

O1 = [1 2 1 | **3 3 1** | 3 1 2 1 1]

O2 = [1 2 2 | **1 2 3** | 1 1 2 1 1]

แต่เนื่องจากเมื่อสลับสตริงย่อย อาจทำให้สตริงคำตอบที่ได้มีจำนวนซ้ำของตัวเลขแต่ละตัวที่ไม่ถูกต้อง ดังนั้นหากมีตำแหน่งที่อยู่นอกสตริงย่อยมีค่าซ้ำกับสตริงย่อยแล้วทำให้จำนวนซ้ำของตัวเลขมากกว่าสตริงพ่อแม่ ให้เปลี่ยนค่าตำแหน่งนั้นเป็น x

จากตัวอย่าง O1 มีเลข 3 เกินมา 1 ตัว และ O2 มี 2 เกินมา 1 ตัว ดังนั้นจึงแทนค่า x แทนตำแหน่งที่เป็นเลข 3 ตัวแรกที่พบใน O1 เป็นจำนวน 1 ตัว และแทนค่า x แทนตำแหน่งที่เป็นเลข 2 ตัวแรกที่พบใน O2 เป็นจำนวน 1 ตัว ได้เป็น

O1 = [1 2 1 | **3 3 1** | x 1 2 1 1]

O2 = [1 X 2 | **1 2 3** | 1 1 2 1 1]

จากนั้นแทนค่า (Map) โดยดูจากค่าในสตริงย่อย นั่นคือ แทนค่า 1 เป็น 3 และ 3 เป็น 1 และแทนค่า 3 เป็น 2 และ 2 เป็น 3

O1 = [1 2 1 | **3 3 1** | x 1 2 1 1]

O2 = [1 X 2 | **1 2 3** | 1 1 2 1 1]

ดังนั้นสตริงที่ได้เมื่อทำการแทนที่แล้วคือ

O1 = [1 2 1 3 3 1 2 1 2 1 1]

O2 = [1 3 2 1 2 3 1 1 2 1 1]

3) วิธี Modified Order Crossover (modOX)

เป็นวิธีการครอสโอเวอร์ที่ดัดแปลงจากวิธี Order Crossover (Davis,1985) เป็นการสุ่มสตริงพ่อแม่ 2 ตัว มาแลกเปลี่ยนสตริงกันเพื่อสร้างเป็นสตริงรุ่นลูก เริ่มต้นโดยการสุ่มตัวเลขขึ้นมา 2 ตัว เพื่อหาสตริงย่อย ตัวเลขสุ่มที่มีค่าน้อยกว่าจะเป็นตำแหน่งเริ่มต้นของสตริงย่อย ส่วนตัวเลขสุ่มที่มีค่ามากจะเป็นตำแหน่งสุดท้ายภายในสตริงย่อย จากนั้นคัดลอกสตริงย่อยจากสตริงพ่อแม่ตัวที่ 1 และ 2 มาใส่ในสตริงลูก ในตำแหน่งเดียวกัน

สมมติว่าทำการสุ่มตัวเลขขึ้นมา 2 ตัวคือ 4 และ 6 ดังนั้นได้ว่าสตริงย่อยที่ได้จากสตริงพ่อแม่ตัวที่ 1(P1) คือ [2 2 3] และสตริงย่อยจากสตริงพ่อแม่ตัวที่ 2 (P2) คือ [1 2 3] จากนั้นคัดลอกสตริงย่อยลงในสตริงลูก (O1 และ O2) ในตำแหน่งเดียวกัน ตำแหน่งที่เหลือเป็นตำแหน่งว่างแทนด้วย *

P1 = [1 1 2 2 2 3 3 3]

O1 = [* * * 2 2 3 * * *]

O2 = [* * * 1 2 3 * * *]

P2 = [1 2 3 1 2 3 2 3 3]

ลบค่าใน P2 ที่ซ้ำกับค่าใน O1 และ ลบค่าใน P1 ที่ซ้ำกับค่าใน O2 โดยเริ่มจากตำแหน่งที่ 1 จนครบจำนวนที่ซ้ำ โดยการแทนที่ด้วย x

P1 = [1 1 2 2 2 3 3 3]

O1 = [* * * 2 2 3 * * *]

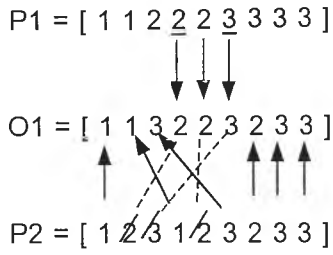
P2 = [1 x x 1 x 3 2 3 3]

P1 = [x 1 x 2 2 x 3 3 3]

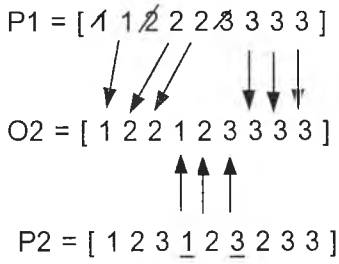
O2 = [* * * 1 2 3 * * *]

P2 = [1 2 3 1 2 3 2 3 3]

นำค่าที่เหลืออยู่ใน P2 มาแทนที่ * ใน O1 ตามลำดับเดิม



และแทนค่าที่เหลือใน P1 มาแทนที่ * ใน O2 ตามลำดับเดิม



ดังนั้นจะได้สตริงรุ่นลูกคือ

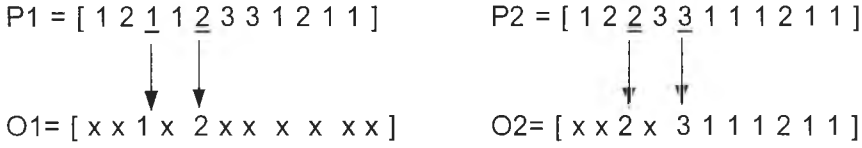
O1 = [1 1 3 2 2 3 2 3 3]

O2 = [1 2 2 1 2 3 3 3 3]

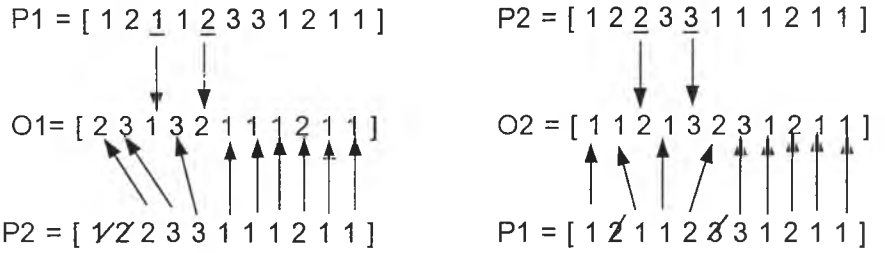
4) วิธี Modified Position Base Crossover (modPBX)

เป็นวิธีที่ดัดแปลงมาจากวิธี Position Base Crossover (Syswerda) โดยเริ่มจากนำสตริงพ่อแม่ที่ได้จากการสุ่ม 2 ตัว จากนั้นเลือกตำแหน่งจากสตริงพ่อแม่อย่างสุ่ม แล้วนำค่าในตำแหน่งที่เลือกของสตริงพ่อแม่ ไปใส่ในตำแหน่งเดียวกันของสตริงลูก

สมมติสุ่มได้ตัวเลข 3 และ 5



จากนั้น ตัดค่าใน P2 ที่ซ้ำกับ O1 แล้วนำค่าที่เหลือใส่ลงใน O1 ในตำแหน่งที่ว่างตามลำดับ และ ตัดค่าใน P1 ที่ซ้ำกับ O2 แล้วนำค่าที่เหลือใส่ลงใน O2 ในตำแหน่งที่ว่างตามลำดับ



ดังนั้นได้ว่าสตริงลูกที่ได้คือ

O1 = [2 3 1 3 2 1 1 1 2 1 1]

O2 = [1 1 2 1 3 2 3 1 2 1 1]

สตริงคำตอบที่ได้จากการครอสโอเวอร์จะเข้าสู่ Mating Pool

6.4.6 การมิวเตชั่น

การมิวเตชั่น (Mutation) เป็นวิธีการพัฒนาสตริงคำตอบวิธีหนึ่ง โดยการสลับตำแหน่งของค่าภายในสตริงคำตอบตัวเดียว ทำให้ได้สตริงคำตอบตัวใหม่เกิดขึ้น

ปัจจัยที่เกี่ยวข้องสำหรับการมิวเตชั่น คือ

- วิธีการมิวเตชั่น (Mutation Type) มีหลายวิธี โดยในงานวิจัยนี้ ได้นำเสนอวิธีการมิวเตชั่น ไว้ทั้งหมด 4 วิธี คือ 1.Inversion Mutation 2.Insertion Mutation 3.Reciprocal Exchange Mutation และ 4.Displacement Mutation
- ความน่าจะเป็นในการมิวเตชั่น (Mutation Probability : *Pm*) คือ ค่าที่ใช้ในการคัดเลือกสตริงที่จะเข้าสู่การมิวเตชั่น โดยใช้เปรียบเทียบกับค่าที่สุ่มขึ้นกับค่า Pm ถ้าสตริงตัวใดที่มีค่าสุ่มน้อยกว่าค่า Pm จะต้องเข้าทำตามกระบวนการมิวเตชั่น
- จำนวนสตริงที่ทำการมิวเตชั่น (*Nm*) คือ จำนวนสตริงที่จะเข้าสู่การมิวเตชั่น

6.4.6.1 การเลือกสตริงคำตอบ

จากสตริงคำตอบจำนวน *popsize* ตัวที่ได้จากผ่านกระบวนการครอสโอเวอร์ จะมีสตริงคำตอบเพียงบางส่วนเท่านั้นที่จะถูกนำมาทำการมิวเตชั่น สตริงคำตอบที่ไม่ได้ถูกคัดเลือกก็จะยังคงสภาพเดิมและอยู่ใน Mating Pool (เป็นประชากรในเจนเนอเรชั่น) ต่อไป

จำนวนสตริงคำตอบที่จะถูกนำมาทำการมิวเตชั่นจับคู่ (Nm) ขึ้นอยู่กับความน่าจะเป็นในการมิวเตชั่น (Pm) โดยการเลือกสตริงคำตอบเพื่อที่จะนำไปมิวเตชั่น มีขั้นตอนดังนี้

1) สร้างตัวเลขสุ่ม r ซึ่งมีค่าระหว่าง 0 ถึง 1 ให้กับสตริงคำตอบแต่ละตัว
2) สตริงคำตอบตัวใดที่ตัวเลขสุ่มมีค่าน้อยกว่า Pm จะถูกเลือกไปทำการมิวเตชั่น

6.4.6.2 การมิวเตชั่น

รายละเอียดการมิวเตชั่นแต่ละวิธีของทั้ง 4 วิธี มีดังนี้

1) วิธี Inversion Mutation

วิธี Inversion เป็นวิธีการมิวเตชั่นโดยการสลับค่าของสตริงย่อยภายในสตริงคำตอบ โดยเริ่มต้นจากการสุ่มตัวเลขสุ่มขึ้นมา 2 ตัว ตัวเลขสุ่มที่มีค่าน้อยกว่าจะเป็นตำแหน่งเริ่มต้นของสตริงคำตอบย่อย ส่วนตัวเลขสุ่มที่มีค่ามากจะเป็นตำแหน่งสุดท้ายภายในสตริงคำตอบย่อย

S = [1 2 2 1 2 3 2 2]

สมมติว่าเราสุ่มตัวเลขขึ้นมา 2 ตัวคือ 3 และ 6 สตริงย่อยที่ใช้ทำการสลับตำแหน่งคือ [2 1 2 3] หรือ

S = [1 2 | 2 1 2 3 | 2 2]

โดยให้ตัวเลขที่อยู่ระหว่าง | แทนสตริงคำตอบย่อย จากนั้นนำสตริงคำตอบย่อยที่ได้มาทำการสลับตำแหน่งโดยให้เรียงลำดับจากขวามาซ้าย ซึ่งตัวเลขตัวสุดท้ายจะกลายเป็นตัวเลขตัวแรก ส่วนตัวเลขตัวแรกก็จะกลายเป็นตัวเลขตัวสุดท้ายของสตริงคำตอบย่อย

S = [1 2 | 2 1 2 3 | 2 2]

↓

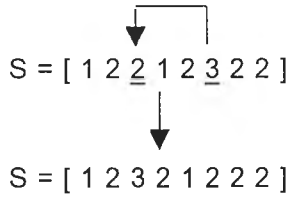
S = [1 2 | 3 2 1 2 | 2 2]

2) วิธี Insertion Mutation

วิธี Insertion เป็นวิธีการย้ายตำแหน่งของตัวเลขภายในสตริงคำตอบ โดยเริ่มจากการสุ่มตัวเลขของตำแหน่งที่ทำการย้ายคำตอบ ซึ่งสมมติให้สุ่มได้เลข 6

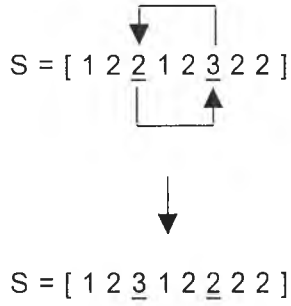
S = [1 2 2 1 2 3 2 2]

จากนั้นทำการสุ่มตัวเลขอีกหนึ่งตัวเพื่อหาตำแหน่งที่จะทำไปแทรกภายในสตริงคำตอบเดียวกัน ซึ่งสมมติให้สุ่มได้เลข 3 เลขที่ได้รับการคัดเลือกในตำแหน่งที่ 6 คือเลข 3 ก็จะไปอยู่แทนที่ในตำแหน่งที่ทำการสุ่ม คือตำแหน่งที่ 3



3) วิธี Reciprocal Exchange Mutation

วิธี Reciprocal Exchange เป็นการสลับตำแหน่งของตัวเลข 2 ตัวภายในสตริงคำตอบ โดยเริ่มจากทำการสุ่มตัว 2 ตัวที่ไม่ซ้ำกัน สมมติให้สุ่มได้เลข 3 และ 6 จากนั้นทำการสลับตำแหน่งของตัวเลขทั้งสอง



4) วิธี Displacement Mutation

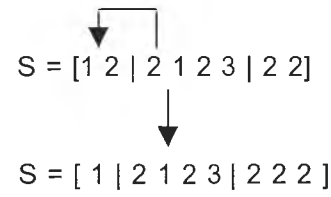
วิธี Displacement เป็นการย้ายตำแหน่งซึ่งคล้ายคลึงกับ วิธี Insertion แต่จะทำการย้ายตำแหน่งของสตริงย่อย ในขั้นแรกจะทำการคัดเลือกสตริงย่อยมาก่อน ซึ่งมีวิธีการคัดเลือกเช่นเดียวกับ วิธี Inversion คือ สุ่มตัวเลขสุ่มขึ้นมา 2 ตัว ตัวเลขสุ่มที่มีค่าน้อยกว่าจะเป็นตำแหน่งเริ่มต้นของสตริงคำตอบย่อย ส่วนตัวเลขสุ่มที่มีค่ามากจะเป็นตำแหน่งสุดท้ายภายในสตริงคำตอบย่อย

S = [1 2 2 1 2 3 2 2]

สมมติว่าสุ่มตัวเลขขึ้นมา 2 ตัวคือ 3 และ 6 สตริงย่อยที่ใช้ทำการสลับตำแหน่งคือ [2 1 2 3] หรือ

S = [1 2 | 2 1 2 3 | 2 2]

โดยให้ตัวเลขที่อยู่ระหว่าง | แทนสตริงคำตอบย่อย จากนั้นสุ่มตัวเลขขึ้นมาอีกหนึ่งตัวเพื่อเลือกตำแหน่งที่จะทำการแทรกสตริงย่อยที่ได้รับการคัดเลือก สมมติว่าได้เลข 2 จากนั้นทำการแทรกสตริงย่อยนี้ลงไปยังตำแหน่งที่ทำการสุ่ม



จะเห็นได้ว่าหากเราทำการสุ่มตัวเลขของสตริงคำตอบย่อยได้ตัวเลขเดียวกัน จะได้ว่าสตริงคำตอบย่อยนั้นเป็นตำแหน่งเดียวกัน ซึ่งการทำมิวเตชั่นจะเป็นการมิวเตชั่นแบบวิธี Insertion (ย้ายตำแหน่งตัวเลขตัวเดียวภายในสตริงคำตอบ)

6.4.7 เทคนิคการเก็บค่าที่ดีที่สุด

เทคนิคการเก็บค่าที่ดีที่สุด (Elite Preserve Strategy) เป็นเทคนิคที่นำมาใช้เพื่อเก็บค่าที่ดีที่สุดไว้ เนื่องจากสตริงคำตอบที่ได้จากการครอสโอเวอร์และการมิวเตชั่น อาจเป็นคำตอบที่แย่กว่าคำตอบที่เคยปรากฏในเจนเนอเรชั่นที่ผ่านๆมา ดังนั้นจึงต้องมีการเก็บค่าที่ดีที่สุด (Elite Preserve Solution) เอาไว้เพื่อใช้เปรียบเทียบกับค่าที่ดีที่สุดของสตริงคำตอบชุดใหม่ ถ้าหาก Elite Preserve Solution ให้ค่า Fitness ที่ดีกว่าค่าที่ดีที่สุดของสตริงชุดใหม่ก็ให้เอา Elite Preserve Solution แทนที่ค่าที่แย่ที่สุด ทั้งนี้เพื่อให้สตริงคำตอบที่ดีที่สุดเท่าที่พบยังคงอยู่ในกระบวนการของ GAs ต่อไป

เทคนิคการเก็บค่าที่ดีที่สุดจะถูกนำไปใช้ระหว่างกระบวนการของ GAs 3 ครั้ง ดังนั้นจึงแบ่งเทคนิคการเก็บค่าที่ดีที่สุดออกเป็น 3 เทคนิค คือ

1) Initial Elite Preserve Strategy

เป็นจุดเริ่มต้นของเทคนิคการเก็บค่าที่ดีที่สุด ซึ่งจะกระทำเพียงครั้งเดียวภายหลังจากการสร้างสตริงคำตอบเบื้องต้น ในตอนต้นของกระบวนการเจนเนติกอัลกอริทึม และผ่านการถอดรหัสรวมทั้งการประเมินค่าเรียบร้อยแล้ว ค่า Fitness ของสตริงแต่ละตัวที่ได้จากการประเมินค่าจะถูกเรียงลำดับจากมากไปน้อย สตริงคำตอบเพียงหนึ่งตัวที่ให้ค่า Fitness มากที่สุดก็จะถูกเลือกไปเป็นคำตอบที่ดีที่สุดที่เก็บไว้ (Elite Preserve Solution) แต่หากมีสตริงคำตอบที่ให้ค่า Fitness ที่เท่ากัน พิจารณาเลือกสตริงคำตอบจากค่า Range Time โดยถ้าสตริงคำตอบตัวใดให้ค่า Range Time ที่ต่ำกว่า จะถูกเลือกเป็นคำตอบที่ดีที่สุดเก็บไว้ จากนั้นสตริงคำตอบทั้งหมดรวมทั้งตัวที่เลือกไปเป็น Elite Preserve Solution จะเข้าสู่ขั้นตอนต่างๆของ GAs ต่อไป

2) Post-crossover Elite Preserve Strategy

เป็นเทคนิคการเก็บค่าที่ดีที่สุดที่ใช้ภายหลังจากที่เสร็จสิ้นกระบวนการครอสโอเวอร์แล้ว ทั้งนี้เนื่องจากว่าสตริงคำตอบที่ได้จากการครอสโอเวอร์อาจเป็นคำตอบที่ดีกว่าคำตอบอื่นๆ ที่เคยพบมา แต่เมื่อนำไปมิวเตชั่นแล้ว สตริงคำตอบตัวนี้จะเปลี่ยนไปและอาจให้คำตอบที่ดีน้อยกว่าเดิม ดังนั้น เพื่อป้องกันไม่ให้สตริงคำตอบที่ดีที่ได้หลังจากการครอสโอเวอร์สูญหายไปจึงต้องทำการการถอดรหัสและประเมินค่าสตริงคำตอบภายหลังการครอสโอเวอร์ทั้งหมด *popsize* ตัว แล้วนำสตริงคำตอบที่ดีที่สุดภายหลังการครอสโอเวอร์ไปเปรียบเทียบกับ Elite Preserve Solution โดยมีเกณฑ์ในการพิจารณาเพื่อเก็บค่าที่ดีที่สุดเป็นดังนี้

- ถ้าสตริงคำตอบภายหลังการครอสโอเวอร์ดีกว่า คือให้ค่า Fitness ที่สูงกว่า ก็ให้เอาสตริงคำตอบที่ดีที่สุดนั้นไปเป็น Elite Preserve Solution แทน
- ถ้า Elite Preserve Solution ดีกว่า ก็ให้นำสตริงคำตอบภายหลังจากการครอสโอเวอร์ทั้งหมด *popsize* ตัวไปผ่านกระบวนการมิวเตชั่นตามปกติ
- ถ้าคำตอบภายหลังจากการครอสโอเวอร์ และ Elite Preserve Solution ให้ค่า Fitness ที่เท่ากัน ก็ให้พิจารณาเลือกสตริงคำตอบจากค่า Range Time โดยหาก Elite Preserve Solution มีค่า Range Time ที่มากกว่าจะเลือกสตริงคำตอบภายหลังการครอสโอเวอร์เป็นคำตอบที่ดีที่สุดเก็บไว้

ยกตัวอย่างเช่น ภายหลังการครอสโอเวอร์ มีสตริงคำตอบ 10 ตัว ซึ่งเมื่อนำไปผ่านกระบวนการถอดรหัสและประเมินค่าแล้วได้ค่า Fitness ของสตริงคำตอบแต่ละตัวเป็น 0.80 0.75 0.82 0.78 0.90 0.85 0.77 0.92 0.95 และ 0.77 จะเห็นว่าค่า Fitness ที่ดีที่สุดใน 10 ตัวนี้คือค่า 0.95 ของสตริงคำตอบตัวที่ 9 ให้เอาค่า 0.95 นี้ไปเปรียบเทียบกับค่า Fitness ของ Elite Preserve Solution หากมีค่าน้อยกว่า 0.95 ก็ให้เอาสตริงคำตอบตัวที่ 9 นี้เป็น Elite Preserve Solution แทนตัวเดิม แต่หากมีค่ามากกว่า0.95 ก็ให้คงคำตอบของ Elite Preserve Solution ไว้ แต่ถ้าให้ค่า Fitness ที่เท่ากัน จะพิจารณาจากค่า Range Time ดังที่กล่าวข้างต้น

หลังจากผ่านขั้นตอนของเทคนิคการเก็บค่าที่ดีที่สุดภายหลังการครอสโอเวอร์แล้ว สตริงคำตอบทั้ง *popsize* ตัว จะเข้าสู่ขั้นตอนการมิวเตชั่นต่อไป

3) Elite Preserve Strategy of Generation

เป็นเทคนิคการเก็บค่าที่ดีที่สุดที่ใช้ภายหลังการมิวเตชั่น ซึ่งถือว่าเป็นการเก็บค่าที่ดีที่สุดของเจนเนอเรชั่นนั้นๆด้วย การเก็บค่าที่ดีที่สุดของเจนเนอเรชั่นจะช่วยให้คำตอบที่ดีที่สุดเท่าที่เคยปรากฏขึ้นมายังคงมีอยู่ในเจนเนอเรชั่นต่อไป การเก็บค่าในขั้นตอนนี้จะทำหลังจากที่มีการมิวเตชั่นเรียบร้อยแล้ว สตริงคำตอบที่ได้ภายหลังการมิวเตชั่นจำนวน *popsize* ตัว จะถูกถอดรหัสและประเมินค่า จากนั้นก็ให้เอาสตริงคำตอบหลังที่ดีที่สุดจากการมิวเตชั่น มาเปรียบเทียบกับ Elite Preserve Solution เช่นเดียวกับในขั้นตอนของ Post-crossver Elite Preserve

Strategy แต่แตกต่างกันตรงที่ จะมีการนำเอา Elite Preserve Solution มาแทนที่คำตอบที่แย่ที่สุดของสตริงคำตอบชุดนี้เมื่อ Elite Preserve Solution เป็นคำตอบที่ดีกว่า

ยกตัวอย่างเช่น ภายหลังการมิวเตชั่นได้สตริงคำตอบ 10 ตัว ที่มีค่า Fitness เป็น 0.77 0.83 0.91 0.80 0.79 0.96 0.78 0.85 0.75 และ 0.74 จะได้ว่าค่า Fitness ที่ดีที่สุดคือ 0.96 ของสตริงคำตอบตัวที่ 6 ซึ่งถ้าค่า Fitness ของ Elite Preserve Solution น้อยกว่า 0.96 สตริงคำตอบตัวที่ 6 จะกลายเป็น Elite Preserve Solution ตัวใหม่ แต่ถ้าค่า Fitness ของ Elite Preserve Solution มากกว่า 0.96 ก็ให้ตัดสตริงคำตอบตัวที่ 10 ซึ่งมีค่า Fitness ต่ำที่สุดทิ้งไป เอาสตริงคำตอบที่เป็น Elite Preserve Solution ขณะนั้นไปใส่แทนที่

สตริงคำตอบที่ได้ภายหลังจากขั้นตอนนี้จะกลายเป็นสตริงคำตอบพ่อแม่ที่แท้จริงในเจนเนอเรชั่นต่อไป

6.5 สรุปท้ายบท

การนำเจนเนติกอัลกอริทึมไปประยุกต์ใช้ในปัญหาการคำตอบที่ดีที่สุด (Optimization Problems) ต่างๆ จะมีโครงสร้างหลักที่คล้ายคลึงกันตามแบบของเจนเนติกอัลกอริทึมอย่างง่าย ประกอบด้วย การเข้ารหัสและสร้างประชากรเบื้องต้น การรีโปรดักชั่น การครอสโอเวอร์ และการมิวเตชั่น แต่รายละเอียดปลีกย่อยในแต่ละปัญหาจะแตกต่างกันตามลักษณะของปัญหา

สำหรับการประยุกต์ใช้ฟัซซี่เจนเนติกอัลกอริทึมสำหรับปัญหาการจัดลำดับงานเข้าสายการประกอบแบบผลิตภัณฑ์ผสมที่มีเวลาการทำงานเป็นฟัซซี่ เป็นการประยุกต์ใช้หลักการฟัซซี่ในขั้นตอนการประเมินค่า เนื่องจากลักษณะปัญหามีเวลาการทำงานเสร็จสิ้นเป็นแบบฟัซซี่ ซึ่งไม่สามารถหาค่าเวลาการทำงานเสร็จสิ้นที่น้อยที่สุดได้โดยตรง จึงมีการกำหนดค่าวัตถุประสงค์เป้าหมาย (Objective Value) คือ ค่าความพึงพอใจของผู้จัดลำดับผลิตภัณฑ์ ซึ่งพิจารณาจากค่า Fitness ที่คำนวณโดยใช้หลักการทางฟัซซี่ โดยลำดับผลิตภัณฑ์ที่มีค่า Fitness สูง แสดงถึงความพึงพอใจของผู้จัดลำดับผลิตภัณฑ์ที่มีความพอใจมาก

สำหรับในงานวิจัยนี้ซึ่งเป็นปัญหาการจัดลำดับผลิตภัณฑ์เข้าสายการประกอบแบบผลิตภัณฑ์ผสมที่มีเวลาการทำงานแบบฟัซซี่ มีการประยุกต์ใช้ฟัซซี่เจนเนติกอัลกอริทึม โดยมีรายละเอียดดังนี้

- การสร้างประชากรเบื้องต้น มีการใส่รหัสคำตอบแบบ Non-binary String ซึ่งมีตัวเลขที่ซ้ำกันได้แสดงถึงชนิดผลิตภัณฑ์ที่เข้าทำการผลิต
- การรีโปรดักชั่น ประกอบด้วย 3 ส่วนย่อยคือ 1.การถอดรหัส ซึ่งเป็นการนำผลิตภัณฑ์ต่างๆในสตริงคำตอบมาทำการผลิตตามลำดับ 2.การประเมินค่า เป็นการคำนวณหาค่าความเหมาะสมของสตริงคำตอบแต่ละตัว และ 3.การคัดเลือกสตริงซึ่งเป็นการคัดเลือกสตริงตัวที่มีความเหมาะสมมากกว่าเพื่อเข้าสู่กระบวนการถัดไปโดยมีทั้งวิธี Roulette Wheel และ Tournament

- การครอสโอเวอร์ วิธีการครอสโอเวอร์ที่ใช้ในงานวิจัยมี 4 วิธี คือ modMOX modPMX modOX และ modPBX ซึ่งเป็นวิธีการที่ปรับปรุงเปลี่ยนแปลงจากวิธีการครอสโอเวอร์แบบเดิม เพื่อให้เหมาะสมกับลักษณะของสตริงคำตอบที่เป็นแบบ Non-binary String ที่มีตัวเลขซ้ำกันได้
- การมิวเตชั่น วิธีการมิวเตชั่นที่นำมาใช้ในงานวิจัยมี 4 วิธี คือ Inversion Insertion Displacement และ Reciprocal Exchange
- เทคนิคการเก็บค่าที่ดีที่สุด เป็นวิธีการที่นำเข้าไปใช้ภายหลังการสตริงคำตอบเบื้องต้น ภายหลังการครอสโอเวอร์ และภายหลังการมิวเตชั่น ทั้งนี้เพื่อให้สตริงคำตอบที่ดียังคงอยู่ต่อไปในเจนเนอเรชั่นถัดไป

สตริงคำตอบที่ได้จากกระบวนการของเจนเนติกอัลกอริทึม จะกลายเป็นประชากรพ่อแม่ในเจนเนอเรชั่นต่อไป และจะเกิดขึ้นซ้ำไปเรื่อยๆ จนกว่าจะถึงเจนเนอเรชั่นสูงสุดที่กำหนด คำตอบที่ต้องการจะได้หลังจากเสร็จสิ้นการทำตามกระบวนการเจนเนติกอัลกอริทึมในเจนเนอเรชั่นสูงสุด