บทที่ 6

การประยุกต์ใช้เจนเนติกอัลกอริทึมกับปัญหาการจัดตารางการผลิตที่มีเวลาของการปรับตั้งเครื่องแบบฟัซซี่ซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า

เจนเนติกอัลกอริทึมถูกนำไปประยุกต์ใช้ในการแก้ปัญหา NP-Hard ต่างๆ เช่น การจัดสมดุลของสายงานการประกอบ (Kim และคณะ, 1996) การวางผังโรงงาน (ชนะ, 1998) รวมทั้งการจัดตารางการผลิต (Murata และคณะ, 1996) ซึ่งมีรูปแบบปัญหาต่างกันออกไป โดยส่วนมากแล้ว เจนเนติกอัลกอริทึมที่ใช้จะมีโครงสร้างหลักคล้ายคลึงกันตามแบบของเจนเนติกอัลกอริทึมอย่างง่าย คือ มีการเข้ารหัสและสร้างประชากรเริ่มต้น (Representation) การรีโปรดักชัน (Reproduction) การครอสโอเวอร์ (Crossover) และการมิวเตชัน (Mutation) แต่อย่างไรก็ตาม เจนเนติกอัลกอริทึมที่ใช้สำหรับแต่ละปัญหาก็จะมีรายละเอียดปลีกย่อยในโครงสร้างหลักที่แตกต่างกันออกไปขึ้นอยู่กับความเหมาะสมต่อรูปแบบปัญหานั้นๆ

เนื้อหาในบทนี้จะนำเสนอการประยุกต์เจนเนติกอัลกอริทึม เพื่อใช้สำหรับปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งเครื่องแบบฟัซซี่ซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า โดยมีกฎเกณฑ์ในการจัด (Criteria) หรือในที่นี้เรียกว่าวัตถุประสงค์ (Objective) เพียงอย่างเดียว คือ เพื่อลดเวลาที่ใช้ในการปรับตั้งเครื่องจักรให้มีค่าน้อยที่สุด

6.1 โครงสร้างและขั้นตอนการทำงานของวิธีการเจนเนติกอัลกอริทึม

โครงสร้างของเจนเนติกอัลกอริทึมเป็นส่วนสำคัญที่กำหนดขั้นตอนในการทำงานต่างๆ เนื้อหาในหัวข้อนี้จึงแบ่งออกเป็น 2 ส่วนคือ โครงสร้างของเจนเนติกอัลกอริทึม ซึ่งจะเป็นกระบวนการหลักในวิธีการเจนเนติกอัลกอริทึม และขั้นตอนของเจนเนติกอัลกอริทึม ซึ่งจะเป็นขั้นตอนย่อยในกระบวนการหลักในวิธีการเจนเนติกอัลกอริทึม

6.1.1 *โครงสร้างหลัก*

โครงสร้างหลักของเจนเนติกอัลกอริทึมเป็นโครงสร้างที่ประกอบไปด้วยกระบวนการต่างๆ ของเจนเนติกอัลกอริทึม โดยทั่วไปประกอบด้วย 5 ส่วนหลักคือ

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

   2.1 Decoding เป็นการแปลความหมายของสตริงคำตอบ ในที่นี้การถอดรหัสคำตอบ

   2.2 Evaluation เป็นการคำนวณหาค่าต่างๆเพื่อนำไปสู่การหาค่าความเหมาะสม (Fitness) ของสตริงคำตอบ

   2.3 Selection เป็นกระบวนการในการคัดเลือกสตริงที่มีความเหมาะสมมากกว่า

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

6.1.2 ขั้นตอนการทำงานของเจนเนติกอัลกอริทึม

ขั้นตอนในการทำงานของกระบวนการเจนเนติกอัลกอริทึมสามารถแบ่งออกเป็นขั้นตอนย่อยๆได้ 12 ขั้นตอนดังนี้

1. **Data Input :** รับข้อมูลเข้าต่างๆ ซึ่งได้แก่ จำนวนของผลิตภัณฑ์ และเวลาที่ใช้ในการปรับตั้งเครื่องจักรของแต่ละผลิตภัณฑ์ (เวลาที่น้อยที่สุด เวลาโดยเฉลี่ย และเวลาที่มากที่สุด)
2. **Representation & Initialization :** นำข้อมูลต่างๆมาสร้างคำตอบเบื้องต้นแบบสุ่มจำนวน *popsize* ตัว โดยผ่านกระบวนการใส่รหัสคำตอบ (Representation) และการสร้างประชากรเบื้องต้น (Initial Population)
3. **Decoding :** นำรหัสคำตอบของประชากรเบื้องต้นทุกตัว มาถอดรหัสคำตอบเพื่อให้ได้คำตอบที่สมบูรณ์และสามารถนำไปคำนวณหาค่าต่างๆที่ต้องการได้
4. **Evaluation :** คำนวณหาค่าที่ต้องการ กล่าวคือ เวลาเฉลี่ยในการปรับตั้งเครื่องจักร จากนั้นนำค่านี้ไปคำนวณหาค่า Fitness ของประชากรเบื้องต้นทุกตัว

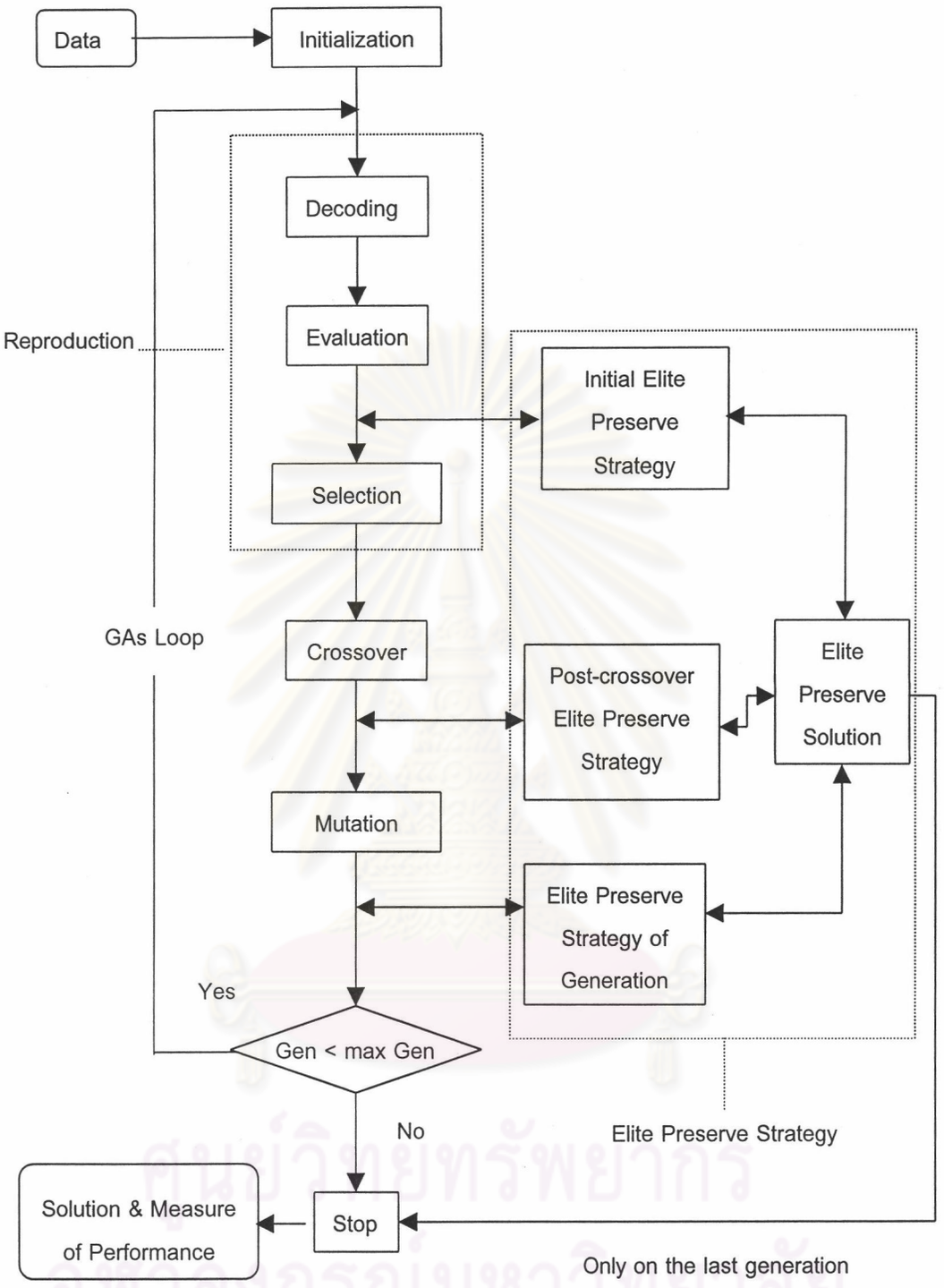
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

5. **Initial Elite Preserve Strategy :** หาค่าคำตอบที่ดีที่สุดจากประชากรเจนเนอเรชันแรก โดยพิจารณาจากเวลาเฉลี่ยและช่วงเวลาในการปรับตั้ง และเก็บค่าคำตอบที่ดีที่สุดนี้ไว้เป็น 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 :** ทำการมิวเตชันสตริงคำตอบที่มีด้วยความน่าจะเป็นในการมิวเตชันเท่ากับ Pm
10. **Elite Preserve Strategy of Generation :** หาค่าคำตอบที่ดีที่สุดจากการมิวเตชันแล้วนำมาเปรียบเทียบกับ Elite Preserve Solution ที่มีอยู่ ถ้าคำตอบที่ได้จากมิวเตชันดีกว่า ก็ให้เก็บคำตอบนั้นเป็น Elite Preserve Solution แทน แต่ถ้า Elite Preserve Solution ดีกว่า ก็ให้แทนที่คำตอบที่แย่ที่สุดจากการมิวเตชันด้วย Elite Preserve Solution
11. **GAs-loop :** ดูว่า เจนเนอเรชัน น้อยกว่าจำนวนเจนเนอเรชันสูงสุดหรือไม่ ถ้าน้อยกว่าให้กลับไปทำข้อที่ 7-10 ถ้าไม่ให้ทำข้อที่ 12
12. **Stop :** หยุดกระบวนการของเจนเนติกอัลกอริทึม และนำค่า Elite Preserve Solution มาเป็นคำตอบ

จากโครงสร้างและขั้นตอนของวิธีการเจนเนติกอัลกอริทึมสามารถเขียนให้อยู่ในรูปของแผนผังการไหล (Flow Chart) ของวิธีการเจนเนติกอัลกอริทึมได้ดังรูปที่ 6.1

6.2 การประยุกต์วิธีการของเจนเนติกอัลกอริทึมสำหรับปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งแบบฟัซซี่ซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า

ในการประยุกต์วิธีการของเจนเนติกอัลกอริทึมสำหรับปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งแบบฟัซซี่ซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า จะเป็นการนำขั้นตอนของวิธีการเจนเนติกอัลกอริทึมต่าง ๆที่ได้กล่าวมาตอนต้นมาประยุกต์เพื่อหาคำตอบที่ดีที่สุดในการจัดตารางการผลิตที่มีเวลาปรับตั้งแบบฟัซซี่ซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า ซึ่งมีขั้นตอนดังนี้คือ



รูปที่ 6.1 *แผนผังแสดงโครงสร้างและวิธีการของเจนเนติกอัลกอริทึม*

6.2.1 การใส่รหัสคำตอบ (Chromosome Representation / Coding)

ขั้นตอนแรกของเจนเนติกอัลกอริทึม คือการกำหนดรูปแบบของการใส่รหัสคำตอบ ซึ่งถือว่าเป็นขั้นตอนที่สำคัญและมีผลอย่างมากต่อขั้นตอนอื่นๆ ของวิธีเจนเนติกอัลกอริ

ทึม การใส่รหัสคำตอบ คือ การเปลี่ยนคำตอบของปัญหาให้อยู่ในรูปของสตริงคำตอบ (หรือที่เรียกว่า Chromosome) เพื่อนำไปพัฒนาคำตอบตามวิธีการของเจนเนติกอัลกอริทึม วิธีการใส่รหัสคำตอบมีทั้งแบบ Binary String คือ การใส่รหัสคำตอบที่ในแต่ละ Bit จะสามารถเลือกใส่รหัสได้ 2 แบบ เช่น 0 หรือ 1 ใช่หรือไม่ใช่ Yหรือ N ฯลฯ และ Non-binary String คือ การใส่รหัสคำตอบที่ในแต่ละ Bit จะสามารถเลือกใส่รหัสได้มากกว่า 2 แบบ เช่น 1-10 ก-ฮ A-Z ฯลฯ ในกรณีของปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งเครื่องแบบฟัซซี่ซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า คำตอบของปัญหาคือการจัดเรียงลำดับในการผลิตของผลิตภัณฑ์ ดังนั้น วิธีการใส่รหัสคำตอบที่ใช้จึงต้องสามารถแสดงลำดับของผลิตภัณฑ์ในรูปแบบของสตริงได้ เพื่อที่จะได้ทราบลำดับการผลิตของแต่ละผลิตภัณฑ์ ซึ่งในลำดับของการผลิตนี้จะมีผลต่อเวลาเฉลี่ยในการปรับตั้งเครื่องจักรรวม วิธีการใส่รหัสคำตอบที่ใช้จึงควรเป็นแบบ Non-binary String

ในปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งเครื่องแบบฟัซซี่ซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า จะมีการจัดตารางการผลิตของผลิตภัณฑ์ต่างๆเหมือนกับจัดตารางการผลิตที่มีเวลาปรับตั้งเครื่องซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า แต่คำตอบที่ได้จะอยู่ในรูปแบบของฟัซซี่ โดยลำดับที่จัดตารางการผลิตจะไม่มีความสัมพันธ์ก่อนหลัง สามารถจัดตารางการผลิตได้อย่างอิสระตามแต่วัตถุประสงค์ที่ต้องการ แต่เวลาของการปรับตั้งเครื่องรวมจะเปลี่ยนไปตามลำดับของการจัดตารางการผลิต จากแนวคิดดังกล่าวลำดับของการจัดตารางการผลิตที่ถูกจัดเรียงสามารถนำมาใช้เป็นรูปแบบของการใส่รหัสคำตอบ โดยลักษณะของสตริงคำตอบมีดังนี้

1) คำตอบ 1 คำตอบ แทนด้วยสตริงคำตอบ 1 ตัว
2) ใน 1 สตริงคำตอบ จะแบ่งเป็นหน่วยเล็กๆที่เรียกว่า Bit เรียงกันอยู่ จำนวนของ Bit จะเท่ากับจำนวนผลิตภัณฑ์ทั้งหมดที่ต้องทำการผลิตบนเครื่องจักรที่พิจารณา
3) ในแต่ละ Bit จะมีค่าตัวเลขตั้งแต่ 1 ถึง m บรรจุอยู่ค่าหนึ่ง ค่านี้หมายถึงหมายเลขที่ใช้แทนผลิตภัณฑ์ชนิดต่างๆ เช่น หมายเลข 1 แทน ผลิตภัณฑ์ชนิดที่ 1
4) ตำแหน่งของ Bit หมายถึงลำดับที่ของผลิตภัณฑ์นั้นๆที่จะทำการผลิต
5) ตัวเลขในแต่ละ Bit ต้องไม่ซ้ำกัน เนื่องจากผลิตภัณฑ์แต่ละชนิดจะถูกผลิตเพียงครั้งเดียวใน 1 รอบ (Cyclic)
6) ลำดับของการผลิตจะเรียงตามตำแหน่งของผลิตภัณฑ์ที่อยู่ใน Bit ก่อนหน้า และเวลาที่ใช้ในการปรับตั้งเครื่องจักรจะคิดจากเวลาในการปรับตั้งระหว่าง Bit แต่ละ Bit

ตัวอย่างเช่น สตริงคำตอบ [1 3 2 5 7 9 8 4 6 10] จะได้ว่า ใน 1 สตริงคำตอบ มี 10 Bit หมายถึง ผลิตภัณฑ์ที่นำมาทำการจัดตารางการผลิตมีทั้งสิ้น 10 ผลิตภัณฑ์ ซึ่งจะทำการผลิตผลิตภัณฑ์ที่ 1 3 2 5 7 9 8 4 6 และ 10 ตามลำดับ

6.2.2 การสร้างกลุ่มประชากรเบื้องต้น (Initial Population Creating)

การสร้างกลุ่มประชากรเบื้องต้น คือ การสร้างคำตอบเบื้องต้นขึ้นมาจำนวนหนึ่ง เพื่อนำไปใช้ในกระบวนการของเจนเนติกอัลกอริทึม โดยสตริงคำตอบ 1 คำตอบ คือ ประชากร 1 ตัว จำนวนของประชากรที่ต้องการสร้างนั้นเป็นพารามิเตอร์ตัวหนึ่งที่ต้องมีการกำหนด ซึ่งในที่นี้กำหนดให้เท่ากับ *popsize* ตัว

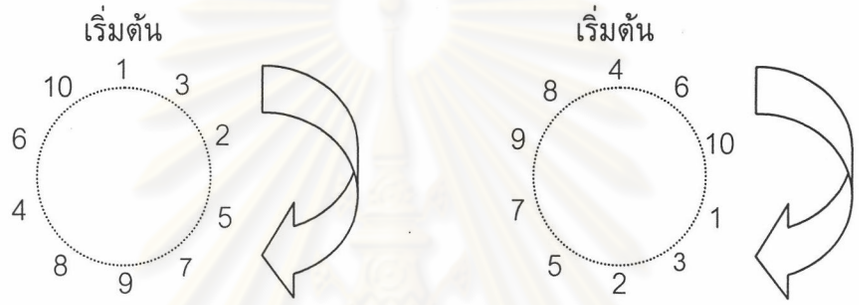
สำหรับปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งเครื่องซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้านั้น ประชากร 1 ตัว หมายถึงลำดับของผลิตภัณฑ์ที่จะทำการผลิตทั้งหมดที่จะทำบนเครื่องจักรนั้น การสร้างประชากรเริ่มต้นตามกระบวนของเจนเนติกอัลกอริทึมจะเป็นการสร้างประชากรแบบสุ่ม หมายถึงตัวเลขที่จะนำมาใส่ในแต่ละ Bit จะต้องถูกเลือกแบบสุ่ม (Random) ดังนั้น การสร้างประชากร 1 ตัวจึงทำได้โดยการใส่ตัวเลขตั้งแต่ 1 ถึง m (m คือจำนวนผลิตภัณฑ์ทั้งหมด) ลงไปในแต่ละ Bit ของสตริงคำตอบโดยจะสุ่มตัวเลขมาใส่ลงในแต่ละ Bit หรือสุ่มลำดับของ Bit ในการใส่ตัวเลขแต่ละตัวจนครบทุก Bit (ตัวเลขใดที่ได้กำหนดลงใน Bit แล้วจะไม่สามารถใส่ซ้ำลงใน Bit อื่นภายในประชากรเดียวกันได้) และทำเช่นนี้ไปเรื่อยๆจนกว่าจะได้ประชากรทั้งหมด *popsize* ตัว

6.2.3 การถอดรหัสคำตอบ (Decoding)

คำตอบที่ปรากฏอยู่ในประชากรหรือสตริงคำตอบที่สร้างขึ้นยังเป็นคำตอบที่ไม่สมบูรณ์ กล่าวคือเป็นเพียงลำดับของผลิตภัณฑ์ที่จะนำไปทำการจัดตารางการผลิตเพียงเท่านั้น ดังนั้นจึงต้องมีการนำผลิตภัณฑ์ตามลำดับที่ได้ในสตริงคำตอบไปจัดให้กับเครื่องจักรที่ทำการผลิตเสียก่อน ซึ่งจะเรียกขั้นตอนนี้ว่าการถอดรหัสคำตอบ (Decoding) การถอดรหัสคำตอบของสตริงคำตอบสำหรับปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งเครื่องซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้านั้น ลำดับในการจัดตารางการผลิตจะคล้ายกับสตริงคำตอบที่สร้างขึ้น แต่เนื่องจากการจัดตารางการผลิตสำหรับปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งเครื่องซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า จะมีลักษณะการทำงานเป็นรอบ เพราะฉะนั้นเมื่อทำการผลิตผลิตภัณฑ์สุดท้ายเสร็จเรียบร้อย ก็จะกลับมาทำการผลิตผลิตภัณฑ์แรกที่ได้จัดตารางการผลิตไว้ ตัวอย่างเช่น สตริงคำตอบ [1 3 2 5 7 9 8 4 6 10] คือเริ่มทำการผลิตผลิตภัณฑ์แรกคือ ผลิตภัณฑ์ชนิดที่ 1 และทำการผลิตผลิตภัณฑ์

ชนิดที่ 3, 2, 5,... จนกระทั่งถึง ผลิตภัณฑ์ชนิดสุดท้ายคือผลิตภัณฑ์ชนิดที่ 10 แล้วจะกลับมาทำการผลิตผลิตภัณฑ์แรกที่ได้จัดตารางการผลิต คือ ผลิตภัณฑ์ชนิดที่ 1 ใหม่ คำตอบที่ได้เมื่อทำการถอดรหัสของสตริงคำตอบ คือ 1-3-2-5-7-9-8-4-6-10-1

อย่างไรก็ดีเนื่องจากปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งเครื่องซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้าจะมีลักษณะของการทำงานที่เป็นรอบ เพราะฉะนั้นจึงมีโอกาสที่จะสร้างกลุ่มประชากรเริ่มต้นแล้วได้ประชากรเดียวกัน ตัวอย่างเช่น สตริงคำตอบ 1-3-2-5-7-9-8-4-6-10 และ สตริงคำตอบ 4-6-10-1-3-2-5-7-9-8 ซึ่งดูเหมือนจะเป็นประชากรคนละตัวกัน แต่เมื่อทำการถอดรหัสคำตอบของสตริงคำตอบจะพบว่าเป็นประชากรตัวเดียวกัน



รูปที่ 6.2 *การนำสตริงคำตอบมาเรียงเป็นวงกลมตามเข็มนาฬิกา*

เมื่อนำคำตอบที่ได้จากการถอดรหัสของสตริงคำตอบทั้ง 2 ตัวดังกล่าว มาเรียงเป็นวงกลมโดยลำดับของผลิตภัณฑ์ที่จัดตารางการผลิตจะมีลำดับการผลิตแบบตามเข็มนาฬิกาดังรูปที่ 6.2 พบว่าสตริงคำตอบทั้ง 2 ตัวเป็นสตริงคำตอบเดียวกัน โดยมีการจัดเรียงลำดับที่เหมือนกัน การสร้างประชากรคำตอบเบื้องต้นจะไม่ยอมให้เกิดคำตอบที่ซ้ำกันขึ้นมา เนื่องจากต้องการประชากรที่หลากหลาย ในการพัฒนาคำตอบตามวิธีการเจนเนติกอัลกอริทึม ดังนั้นภายหลังจากการถอดรหัสคำตอบแล้ว จึงต้องมีการตรวจสอบดูว่าผลการถอดรหัสที่ได้เหมือนกันหรือไม่ ถ้าเหมือนกันก็ให้สร้างประชากรเบื้องต้นตัวใหม่ที่ไม่ซ้ำกับตัวเดิมขึ้นมาแทน

ในกรณีที่เป็นการถอดรหัสคำตอบของสตริงคำตอบที่ได้จากการครอสโอเวอร์ หรือการมิวเตชัน ไม่จำเป็นต้องพิจารณาผลการถอดรหัสว่าให้คำตอบที่เหมือนกันหรือไม่ ทั้งนี้เนื่องจากในขั้นตอนของการครอสโอเวอร์ และมิวเตชันจะยอมให้เกิดคำตอบที่ซ้ำกันขึ้นมาได้

การถอดรหัส ไม่เพียงแต่ให้คำตอบว่าจะจัดตารางการผลิตอย่างไร แต่ยังให้ค่าของเวลาการปรับตั้งของเครื่องจักรรวม (เวลาที่น้อยที่สุด เวลาเฉลี่ย และเวลาที่มากที่สุด) และช่วงเวลาในการปรับตั้งเครื่องจักรอีกด้วย

6.2.4 การประเมินค่า (Evaluation)

ก่อนที่จะเข้าสู่ขั้นตอนการคัดเลือกของเจนเนติกอัลกอริทึม จำเป็นที่จะต้องมีการประเมินค่าประชากรแต่ละตัวเสียก่อนว่ามีความเหมาะสมมากหรือน้อยเพียงใด ความเหมาะสมนี้จะวัดจากค่า Fitness ของสตริงคำตอบแต่ละตัว ตัวใดที่มีค่า Fitness มากก็หมายความว่ามีความเหมาะสมมากตามไปด้วย โดยที่ค่า Fitness ดังกล่าวหมายถึง ค่าของฟังก์ชันวัตถุประสงค์ที่ต้องการทำให้ต่ำที่สุดหรือสูงที่สุดนั่นเอง

ค่าฟังก์ชันวัตถุประสงค์ในที่นี้คือ เวลาเฉลี่ยในการปรับตั้งเครื่องจักรรวม ที่ต้องการทำให้มีค่าต่ำที่สุดนั่นเอง การหาค่าเวลาเฉลี่ยของการปรับตั้งเครื่องรวมของสตริงแต่ละตัวสามารถหาได้จากสมการที่ 6.1

$$X_i = \sum_{j=1}^{m-1} T_{mean}(C_{(j)}, C_{(j+1)}) + T_{mean}(C_{(m)}, C_{(1)}) \qquad (6.1)$$

เมื่อ X_i คือ เวลาเฉลี่ยในการปรับตั้งเครื่องจักรรวมของแต่ละสตริงคำตอบ

i คือ ลำดับที่ของประชากรเริ่มต้น (i=1,2,3,...,$popsize$)

m คือ จำนวนของผลิตภัณฑ์ทั้งหมดที่ทำการจัดตารางการผลิต

j คือ ลำดับของผลิตภัณฑ์ในการจัดตารางการผลิต (j=1,2,3,...,m)

T_{mean} คือ เวลาเฉลี่ยในการปรับตั้งเครื่องจักรของแต่ละผลิตภัณฑ์

C คือ ผลิตภัณฑ์ที่อยู่ในตารางการผลิต C ตัวที่อยู่ข้างหน้าคือผลิตภัณฑ์ที่ผลิตเสร็จแล้ว ส่วน C ตัวที่อยู่ข้างหลังคือผลิตภัณฑ์ที่เริ่มทำการผลิต

สมการดังกล่าวอยู่ในรูปแบบของการหาค่าต่ำที่สุด ดังนั้นต้องทำการแปลงให้อยู่ในรูปแบบการหาค่าสูงที่สุดของ Fitness Function เสียก่อน ดังสมการที่ 6.2

$$f(X_i) = \left(\sum_{i=1}^{popsize} X_i \right) - X_i \qquad (6.2)$$

เมื่อ $f(X_i)$ คือ Fitness Function ของสตริงคำตอบแต่ละตัว

$popsize$ คือ จำนวนประชากรทั้งหมด

นอกเหนือไปจากการคำนวณหาเวลาเฉลี่ยในการปรับตั้งเครื่องจักรรวม และค่า Fitness ของสตริงคำตอบแต่ละตัวแล้ว ในขั้นตอนของการประเมินค่ายังสามารถหาค่าอื่นๆ ที่เกี่ยวข้อง คือ

ก. เวลาที่น้อยที่สุดที่ใช้ในการปรับตั้งเครื่องจักรรวม คือเวลารวมของการใช้เวลาปรับตั้งเครื่องจักรที่น้อยที่สุดของแต่ละผลิตภัณฑ์ เนื่องจากรูปแบบของปัญหาเป็นปัญหาที่มีเวลาการปรับตั้งเครื่องจักรแบบฟัซซี่ เพราะฉะนั้นเวลาที่ใช้ในการปรับตั้งเครื่องจักรในแต่ละรอบการผลิตจะไม่เท่ากัน โดยจะทราบเวลาเพียงคร่าวๆในการใช้ปรับตั้งเครื่องจักร ซึ่งจะมีเวลาการปรับตั้งน้อยที่สุด เวลาปรับการตั้งโดยเฉลี่ย และเวลาการปรับตั้งที่มากที่สุด โดยเวลาการปรับตั้งที่น้อยที่สุดและมากที่สุดนั้นจะนำไปคำนวณหาช่วงเวลาในการปรับตั้งรวม เพื่อที่จะดูโอกาสของเวลาการปรับตั้งที่จะเกิดการเบี่ยงเบนจากค่าเฉลี่ย และใช้ในการพิจารณาเพื่อที่จะเลือกลำดับของการจัดตารางการผลิตในกรณีที่เวลาเฉลี่ยการปรับตั้งเครื่องจักรรวมเท่ากัน แต่ลำดับในการจัดตารางการผลิตไม่เหมือนกัน โดยสามารถหาเวลาที่น้อยที่สุดการปรับตั้งเครื่องจักรได้ดังสมการที่ 6.3

$$Y_i = \sum_{j=1}^{m-1} T_{\min}(C_{(j)}, C_{(j+1)}) + T_{\min}(C_{(m)}, C_{(1)}) \qquad (6.3)$$

เมื่อ Y_i คือ เวลาที่น้อยที่สุดในการปรับตั้งเครื่องจักรรวมของแต่ละสตริงคำตอบ

$T_{\min}$ คือ เวลาที่น้อยที่สุดในการปรับตั้งเครื่องจักรของแต่ละผลิตภัณฑ์

ข. เวลาที่มากที่สุดที่ใช้ในการปรับตั้งเครื่องจักรรวม คือเวลารวมของการใช้เวลาปรับตั้งเครื่องจักรที่มากที่สุดของแต่ละผลิตภัณฑ์ ค่าที่ได้นี้จะนำไปคำนวณหาช่วงเวลาในการปรับตั้งรวมเช่นเดียวกันกับเวลาที่น้อยที่สุดที่ใช้ในการปรับตั้งเครื่องจักรรวม ซึ่งสามารถหาค่านี้ได้จากสมการที่ 6.4

$$Z_i = \sum_{j=1}^{m-1} T_{\max}(C_{(j)}, C_{(j+1)}) + T_{\max}(C_{(m)}, C_{(1)}) \qquad (6.4)$$

เมื่อ Z_i คือ เวลาที่มากที่สุดในการปรับตั้งเครื่องจักรรวมของแต่ละสตริงคำตอบ

$T_{\max}$ คือ เวลาที่มากที่สุดในการปรับตั้งเครื่องจักรของแต่ละผลิตภัณฑ์

ค. ช่วงเวลาในการปรับตั้งเครื่องจักรรวม คือผลต่างของเวลาการปรับตั้งเครื่องจักรที่น้อยที่สุดและมากที่สุด ซึ่งสามารถบอกถึงช่วงของเวลาในการปรับตั้งเครื่องจักรได้ และเป็นตัวที่ช่วยพิจารณาการเลือกคำตอบที่ดีกว่าในจัดตารางการผลิต

ในกรณีที่เวลาเฉลี่ยในการปรับตั้งเครื่องจักรรวมเท่ากัน แต่ลำดับในการจัดตารางต่างกัน โดยสตริงตัวใดที่มีช่วงเวลาในการปรับตั้งน้อยกว่าจะได้รับการพิจารณาในการเลือกให้เป็นคำตอบที่ดีกว่า ซึ่งช่วงเวลาในการปรับตั้งเครื่องจักรรวมสามารถหาได้จากสมการที่ 6.5

$$Range\ (I) = Z_i - Y_i \qquad (6.5)$$

เมื่อ *Range (I)* คือ เวลาในการปรับตั้งเครื่องจักรรวมของแต่ละสตริงคำตอบ

6.2.5 การคัดเลือกคำตอบ (Selection)

การคัดเลือกคำตอบทำโดยนำเอากลุ่มสตริงคำตอบเบื้องต้นทั้งหมดมาผ่านวิธีการคัดเลือกโดยดูจากค่า Fitness ของสตริงคำตอบแต่ละตัวเป็นหลัก สตริงตำตอบตัวที่มีค่า Fitness มากก็มีโอกาสที่จะถูกคัดเลือกไว้มากกว่าตัวที่มีค่า Fitness น้อย สตริงคำตอบที่ผ่านการคัดเลือกจำนวน *popsize* ตัวจะผ่านเข้าสู่ Mating Pool เพื่อรอการจับคู่และการดำเนินการของเจนเนติกอัลกอริทึม ในขั้นต่อไป

การคัดเลือกคำตอบที่ใช้มี 2 วิธี คือ วิธี Roulette Wheel Selection (Holland, 1975) และวิธี Tournament Selection (Goldberg, 1991) ซึ่งเป็นวิธีที่ดัดแปลงมาจากวิธี Roulette Wheel Selection

6.2.5.1 วิธี Roulette Wheel Selection

วิธี *Roulette Wheel Selection* เป็นวิธีการคัดเลือกสตริงโดยการสร้างวงล้อรูลเล็ทขึ้นมา แล้วสุ่มตัวเลขในการคัดเลือกสตริงคำตอบ หากตัวเลขนั้นตรงกับส่วนใดของวงล้อรูลเล็ท สตริงคำตอบที่อยู่ภายใต้พื้นที่ของวงล้อรูลเล็ทนั้นจะได้รับการคัดเลือกให้คงอยู่ต่อไป

วงล้อรูลเล็ท (Roulette Wheel) คือ วงกลมที่มีพื้นที่ขนาด 1 หน่วยซึ่งพื้นที่ถูกแบ่งออกเป็นส่วนๆตามจำนวนของประชากรในแต่ละเจนเนอเรชัน (เท่ากับ *popsize* ส่วน) พื้นที่แต่ละส่วนจะมีขนาดเท่ากับความน่าจะเป็นในการถูกเลือกของสตริงคำตอบแต่ละตัว ซึ่งมีวิธีการสร้างและวิธีการเลือกสตริงคำตอบดังนี้

1. หาค่า Fitness รวมของสตริงคำตอบทั้งหมด *popsize* ตัว ดังสมการที่ 6.6

$$F = \sum_{i=1}^{popsize} f(X_i) \qquad (6.6)$$

เมื่อ F คือ ค่า Fitness รวมของสตริงคำตอบทั้งหมด

$f(X_i)$ คือ ค่า Fitness ของสตริงตัวที่ i

2. หาค่าความน่าจะเป็นในการถูกคัดเลือก (Probability of Selection) ของสตริงคำตอบแต่ละตัว ตามสมการที่ 6.7

$$p_i = \frac{f(X_i)}{F} \qquad i = 1, 2, \ldots, popsize \qquad (6.7)$$

เมื่อ p_i คือ ค่าความน่าจะเป็นในการถูกคัดเลือก

3. หาค่าความน่าจะเป็นในการถูกคัดเลือกสะสม (Cumulative Probability of Selection) ของสตริงคำตอบแต่ละตัว ตามสมการที่ 6.8

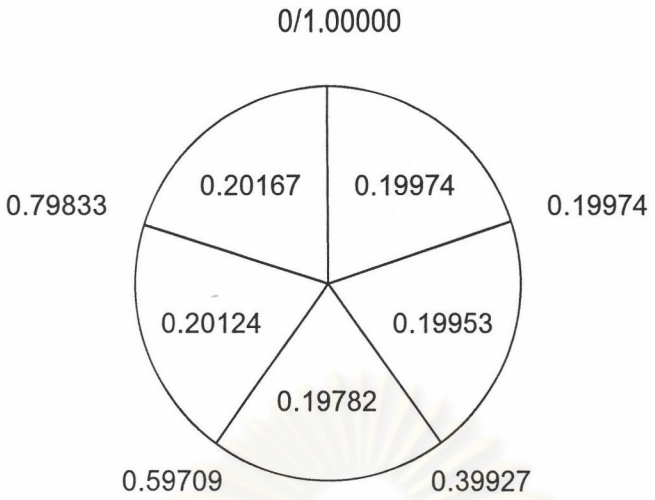
$$q_i = \sum_{j=1}^{i} p_j \qquad (6.8)$$

เมื่อ q_i คือ ค่าความน่าจะเป็นในการถูกคัดเลือกสะสม

ตัวอย่างของวงล้อรูลเล็ทแสดงได้ดังตารางที่ 6.1 และรูปที่ 6.4

ตารางที่ 6.1 *ตัวอย่างตารางแสดงการสร้างวงล้อรูลเล็ท*

String No.	Fitness	p_i	q_i
1	934	0.19974	0.19974
2	933	0.19953	0.39927
3	925	0.19782	0.59709
4	941	0.20124	0.79833
5	943	0.20167	1.000
รวม	4676	1.000	



***รูปที่ 6.3** วงล้อรูลเล็ท*

4. ทำการสร้างตัวเลขสุ่ม r ซึ่งมีค่าระหว่าง 0 ถึง 1 ขึ้นมา 1 ค่า คือ r_1
5. ถ้า $r_1 < q_1$ ให้เลือกสตริงคำตอบตัวที่ 1 แต่ถ้า $q_{i-1} < r_1 < q_i$ (เมื่อ $2< i <$ *popsize*) ให้เลือกสตริงคำตอบตัวที่ i เข้าสู่ Mating Pool ตัวอย่างเช่น หากตัวเลขสุ่ม r_1 มีค่าเท่ากับ 0.118 ก็นำมาเทียบกับค่า q_i ปรากฏว่ามีค่าไม่เกินค่า q_1 (0.200) ก็จะเลือกสตริงคำตอบตัวที่ 1 (String No.1) เข้าสู่ Mating Pool แต่หากตัวเลขสุ่ม r_1 มีค่าเท่ากับ 0.637 แล้วเมื่อนำมาเทียบกับค่า q_i จะพบว่าอยู่ในค่าระหว่าง q_2 และ q_3 ($q_2 < r_1 < q_3$) ดังนั้นจึงเลือกสตริงคำตอบตัวที่ 3 (String No. 3) เข้าสู่ Mating Pool
6. ทำตามขั้นตอนข้อที่ 4 และ 5 จนกว่าจะได้สตริงคำตอบใน Mating Pool ครบ *popsize* ตัว

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

6.2.5.2 วิธี Tournament Selection

การคัดเลือกสตริงคำตอบโดยวิธี Roulette Wheel Selection ซึ่งใช้กันอยู่ทั่วไป จะใช้สุ่มสตริงคำตอบจากวงล้อรูลเล็ท ซึ่งมีโอกาสที่จะสุ่มได้สตริงคำตอบที่มีค่า Fitness น้อยๆด้วย แต่สำหรับการคัดเลือกสตริงคำตอบโดยวิธี Tournament Selection เป็นการสุ่มสตริงคำตอบจากวงล้อรูลเล็ทมา 2 ตัว แล้วนำค่า Fitness มา

เปรียบเทียบกันอีกครั้งหนึ่ง สตริงคำตอบที่ถูกเลือกจึงเป็นตัวที่มีความเหมาะสมมากกว่า สำหรับขั้นตอนการเลือกมีดังนี้

1. สร้างตัวเลขสุ่ม r ซึ่งมีค่าระหว่าง 0 ถึง 1 ขึ้นมา 1 ค่า คือ r_1
2. ถ้า $r_1 < q_1$ ให้เลือกสตริงคำตอบตัวแรก แต่ถ้า $q_{i-1} < r_1 < q_i$ (เมื่อ $2< i <$ *popsize*) ให้เลือกสตริงคำตอบตัวที่ i มาเป็นสตริงคำตอบตัวแรก
3. สร้างตัวเลขสุ่ม r ซึ่งมีค่าระหว่าง 0 ถึง 1 ขึ้นมาอีก 1 ค่า คือ r_2
4. ถ้า $r_2 < q_1$ ให้เลือกสตริงคำตอบตัวแรก แต่ถ้า $q_{i-1} < r2 < q_i$ (เมื่อ $2< i <$ *popsize*) ให้เลือกสตริงคำตอบตัวที่ i มาเป็นสตริงคำตอบตัวที่สอง
5. 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
6. ทำตามขั้นตอนข้อที่ 1 – 5 จนกว่าจะได้สตริงคำตอบใน Mating Pool ครบ *popsize* ตัว

ตารางที่ 6.2 แสดงตัวอย่างการคัดเลือกด้วยวิธี Tournament Selection ซึ่งจะเห็นได้ว่าสตริงคำตอบหมายเลข 5 ซึ่งมีค่า Fitness มากที่สุด จะถูกสุ่มเลือกขึ้นมามากครั้งที่สุด ในขณะที่สตริงคำตอบซึ่งมีค่า Fitness น้อยก็จะถูกสุ่มเลือกน้อยครั้งเช่นกัน ข้อสังเกตประการหนึ่งจากตัวอย่างก็คือ ในการสุ่ม ก็สุ่มได้สตริงคำตอบหมายเลข 3 ที่มีค่า Fitness น้อยที่สุดด้วย ซึ่งถ้าใช้วิธี Roulette Wheel สตริงคำตอบหมายเลข 3 นี้ก็จะมีโอกาสที่จะถูกเลือกเข้าสู่ Mating Pool และจะได้รับการดำเนินการตามกระบวนการเจนเนติกอัลกอริทึมต่อไป แม้ว่าสตริงคำตอบตัวนี้จะมีความเหมาะสมต่ำก็ตาม แต่เมื่อใช้วิธีคัดเลือกแบบ Tournament Selection สตริงคำตอบหมายเลข 3 นี้จะต้องถูกนำไปเปรียบเทียบกับสตริงคำตอบอีกตัวก่อน ดังนั้นโอกาสที่สตริงคำตอบตัวนี้จะถูกเลือกเข้าสู่ Mating Pool ก็จะลดลง

ตารางที่ 6.2 *ตัวอย่างการคัดเลือกด้วยวิธี Tournament Selection*

ครั้งที่	ประชากรตัวที่ 1				ประชากรตัวที่ 2				หมายเลขประชากรที่เลือก
	r_1	$qi > r_1$	หมายเลขประชากร	ค่า Fitness	r_2	$qi > r_2$	หมายเลขประชากร	ค่า Fitness	
1	0.801	1.0000	5	0.2017	0.350	0.3993	2	0.1995	5
2	0.382	0.3993	2	0.1995	0.198	0.1997	1	0.1997	1
3	0.179	0.1997	1	0.1997	0.215	0.3993	2	0.1995	1
4	0.984	1.0000	5	0.2017	0.468	0.5971	3	0.1978	5
5	0.685	0.7983	4	0.2012	0.975	1.0000	5	0.2017	5

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

อย่างไรก็ดี แต่ละวิธีการคัดเลือกสตริงคำตอบจะมีข้อดี-เสียแตกต่างกัน เช่น วิธี Roulette Wheel Selection จะมีข้อดีคือ การได้สตริงคำตอบที่หลากหลายเพื่อนำไปใช้ในการพัฒนาคำตอบ แต่จะมีข้อเสียคือ สตริงคำตอบที่มีค่า Fitness น้อยจะถูกคัดเข้าไปสู่ Mating Pool ซึ่งอาจจะทำให้คำตอบที่ได้ทำการพัฒนาจากค่าของสตริงเริ่มต้น มีโอกาสที่จะได้ค่าที่ดีขึ้นหรือแย่ลงก็ได้ ส่วนวิธี Tournament Selection จะมีข้อดีคือ สตริงที่ถูกคัดเลือกจะเป็นสตริงคำตอบที่ดีกว่าทำให้โอกาสในการพัฒนาคำตอบที่ดีขึ้นเป็นไปได้สูงเสมือนเป็นการพัฒนาคำตอบไปในตัว แต่จะมีข้อเสียคือคำตอบที่ได้อาจจะติดอยู่ใน Local Optimal ซึ่งเป็นคำตอบที่ดีแต่ไม่ใช่คำตอบที่ดีที่สุด เนื่องจากโอกาสที่จะคัดสตริงตัวเดียวกันเข้าไปสู่ Mating Pool จะมีสูงทำให้การ Crossover มีประสิทธิภาพที่ลดลง หรืออาจจะไม่มีผลหากมีสตริงตัวเดียวกันอยู่ใน Mating Pool ทั้งหมด ดังนั้นในการเลือกวิธีการคัดเลือกคำตอบควรมีการทดสอบเพื่อเปรียบเทียบ ทั้ง 2 สองวิธี เพื่อให้เหมาะสมกับการนำไปประยุกต์ใช้แก้ปัญหาต่างๆในแต่ละประเภท

6.2.6 การครอสโอเวอร์ (Crossover)

การ Crossover ถือได้ว่าเป็นส่วนหนึ่งในวิธีการพัฒนาคำตอบของเจนเนติกอัลกอริทึม ซึ่งเป็นการนำเอาสตริงคำตอบมาจับเป็นคู่ๆ แล้วจึงนำ Bit มาสลับกันในแต่ละคู่ การครอสโอเวอร์เป็นขั้นตอนที่กระทำภายหลังการรีโปรดักชัน สตริงที่ถูกเลือกมาทั้งหมดที่อยู่ใน Matting pool จะถูกกำหนดค่าสุ่มให้ สตริงตัวใดที่มีค่าสุ่มน้อยกว่าค่าความน่าจะเป็นในการครอสโอเวอร์ (Pc) จะถูกนำไปจับคู่ กลายเป็นสตริงพ่อแม่ (Parent) แล้วมีการแลกเปลี่ยนบางส่วนของสตริงคำตอบทั้งสองเพื่อสร้างสตริงคำตอบรุ่นลูก (Offspring) ซึ่งจะกระบวนการทำ Crossover จะมีด้วยกันหลายวิธี คือ Modified One-point Crossover

(MOX), Partial Match Crossover (PMX), Cycle Crossover (CX), Order Crossover (OX) และ Position-Base Method (PMX) โดยวิธีการทำ Crossover ดังนี้

6.2.6.1 การจับคู่สตริงคำตอบ

จากสตริงคำตอบจำนวน *popsize* ตัวที่ได้มาจากกระบวนการคัดเลือก จะมีสตริงคำตอบเพียงบางส่วนเท่านั้นที่จะถูกนำมาจับคู่เพื่อเตรียมสำหรับกระบวนการครอสโอเวอร์ สตริงคำตอบที่ไม่ได้ถูกนำไปจับคู่ก็จะยังคงสภาพเดิมและอยู่ใน Mating Pool (เป็นประชากรในเจนเนอเรชัน) 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

6.2.6.2 การครอสโอเวอร์

สตริงคำตอบที่เตรียมไว้ $Nc/2$ คู่จะถูกนำมาผ่านกระบวนการครอสโอเวอร์ ซึ่งเป็นกระบวนการที่นำสตริงคำตอบที่ถูกจับคู่ไว้มาแลกเปลี่ยนส่วนซึ่งกันและกัน

เพื่อให้เกิดสตริงใหม่ขึ้นโดยจะเรียกสตริงคำตอบ 2 ตัวที่ถูกจับคู่นี้ว่า *"สตริงคำตอบรุ่นพ่อแม่ (Parent)"* และจะเรียกสตริงคำตอบ 2 ตัวที่ได้จากการครอสโอเวอร์นี้ว่า *"สตริงคำตอบรุ่นลูก (Offspring)"* วิธีการครอสโอเวอร์มีหลายวิธี ในที่นี้เสนอวิธีการครอสโอเวอร์ไว้ทั้งหมด 5 วิธี คือ

1) วิธี Modified One-point Crossover (MOX)

วิธีการครอสโอเวอร์แบบตำแหน่งเดียวทำได้โดยการหาจุดครอสโอเวอร์ที่สตริงคำตอบมา 1 ตำแหน่ง แบ่งสตริงคำตอบออกเป็น 2 ส่วน แล้วนำส่วนท้ายของสตริงตัวที่แบ่งแล้วตัวหนึ่งต่อเข้ากับส่วนหัวของสตริงตัวที่แบ่งแล้วอีกตัวหนึ่ง ผลการครอสโอเวอร์จะได้สตริงคำตอบรุ่นลูกจำนวน 2 ตัว แต่วิธีนี้อาจทำให้สตริงมีลักษณะที่ไม่ถูกต้อง กล่าวคืออาจทำให้ *gene* บางตำแหน่งเกิดค่าซ้ำกันซึ่งอาจทำให้เกิดการจัดตารางการผลิตของงานที่ซ้ำกันได้ ดังนั้นจึงได้ดัดแปลงวิธีการครอสโอเวอร์แบบตำแหน่งเดียวเพื่อป้องกันการเกิดค่าซ้ำ โดยการครอสโอเวอร์วิธีนี้จะเริ่มจากการกำหนดจุดครอสโอเวอร์ (Crossover Point: Xp) ขึ้นมา 1 ตำแหน่งโดยการสุ่มค่า ระหว่าง [1,m - 1] โดยที่ m คือความยาวของสตริง ตำแหน่งการครอสโอเวอร์แทนด้วย " | " และกำหนดให้ตำแหน่ง Bit ที่ 1 ถึง Xp ของ Parent ตัวที่ 1 เป็น H1 ตำแหน่ง Bit ที่ 1 ถึง Xp ของ Parent ตัวที่ 2 เป็น H2

$$p_1 = [\underbrace{9\ 8\ 4}_{H1} \mid 5\ 6\ 7\ 1\ 3\ 2\ 10\ ]$$

$$p_2 = [\underbrace{8\ 7\ 1}_{H2} \mid 2\ 3\ 10\ 9\ 5\ 4\ 6\ ]$$

นำ H1 มาเป็นส่วนหัวของ Offspring ตัวที่ 1 และนำ H2 มาเป็นส่วนหัวของ Offspring ตัวที่ 2 ตำแหน่งที่เหลือเป็นตำแหน่งว่างแทนด้วย #

$$o_1 = [\ 9\ 8\ 4 \mid \#\ \#\ \#\ \#\ \#\ \#\ \#\ ]$$

$$o_2 = [\ 8\ 7\ 1 \mid \#\ \#\ \#\ \#\ \#\ \#\ \#\ ]$$

ลบค่าใน p_1 ที่ซ้ำกับค่าใน o_2 ออกโดยการแทน x ลงในตำแหน่งที่ซ้ำใน p_2 และ ลบค่าใน p_2 ที่ซ้ำกับค่าใน o_1 ออกโดยการแทน x ลงในตำแหน่งที่ซ้ำใน p_1

o_2 = [8 7 1 | # # # # # # #]

p_1 = [9 x 4 5 6 x x 3 2 10]

o_1 = [9 8 4 | # # # # # # #]

p_2 = [x 7 1 | 2 3 10 x 5 x 6]

นำค่าที่เหลืออยู่ใน p_1 มาแทนที่ # ใน o_2 ตามลำดับเดิม และนำค่าที่เหลืออยู่ใน p_2 มาแทนที่ # ใน o_1 ตามลำดับเดิมเช่นกัน ก็จะได้สตริงคำตอบรุ่นลูกที่สมบูรณ์

o_1 = [9 8 4 7 1 2 3 10 5 6]

o_2 = [8 7 1 9 4 5 6 3 2 10]

2) วิธี Partial Match Crossover (PMX)

วิธีการของ PMX อย่างแรกคือการเลือกคู่สตริงพ่อแม่ขึ้นมาอย่างสุ่ม จากนั้นทำการเลือกตำแหน่งที่จะทำการครอสโอเวอร์อย่างสุ่ม (มีจุดที่ทำการครอสโอเวอร์ 2 จุด ภายในสตริงคำตอบเดียวกัน) ที่อยู่ในช่วง [1,m - 1] โดยที่ m คือความยาวของสตริง และ Bit ตัวแรกคือ Bit หมายเลข 1 และ Bit ตัวสุดท้ายคือ Bit หมายเลข m ขอบเขตของการครอสโอเวอร์อยู่ในระหว่างเครื่องหมาย " | "

p_1 = [9 8 4 | 5 6 7 | 1 3 2 10]

p_2 = [8 7 1 | 2 3 10 | 9 5 4 6]

ในขั้นตอนต่อไปจะทำการสลับค่าระหว่างสตริงที่อยู่ในช่วง " | " นั่นคือตำแหน่งสุ่มอยู่ในช่วง [4,6] ของโครโมโซมลูกหลานทั้งสอง โดยที่ค่าที่อยู่นอกเครื่องหมาย " | " และเป็นค่าที่ซ้ำกันกับค่าที่อยู่ในช่วงที่กำหนดให้เป็น x

จะเห็นได้ว่าตำแหน่งที่สองของสตริง p_2 มีค่าเป็น 7 ซึ่งซ้ำกับค่าที่อยู่ในช่วง " | " ของสตริง p_1 ก่อนทำการสลับ จึงเปลี่ยนค่าที่ซ้ำให้เป็น x ก่อนและตำแหน่งอื่นๆที่มีค่าซ้ำกันก็จะเปลี่ยนเป็น # ด้วยเช่นกัน

o_1 = [9 8 4 | 2 3 10 | 1 # # #]

o_2 = [8 x 1 | 5 6 7 | 9 x 4 x]

แล้วทำการแทนค่า (Map) ดังต่อไปนี้ 2 เป็น 5 และ 5 เป็น 2; 3 เป็น 6 และ 6 เป็น 3; 0 เป็น 7 และ 7 เป็น 0 การแทนค่าเหล่านี้ได้มาจากค่าที่อยู่ในช่วง " | " โดยพิจารณาตำแหน่งของสตริงที่ตรงกัน สตริงที่ได้เมื่อทำการแทนที่แล้วคือ

o_1 = [9 8 4 2 3 10 1 6 5 7]
o_2 = [8 10 1 5 6 7 9 2 4 3]

3) วิธี Cycle crossover (CX)

ขั้นตอนแรกเลือกคู่สตริงพ่อแม่ขึ้นมาอย่างสุ่ม พิจารณาดังต่อไปนี้

p_1 = [2 10 3 4 8 5 9 1 7 6]
p_2 = [5 9 2 1 10 6 7 3 8 4]

พิจารณาที่ตำแหน่งแรกของสตริง p_1 และ p_2 คงเดิมไว้

o_1 = [2 - - - - - - - - -]
o_2 = [5 - - - - - - - - -]

ค่าตำแหน่งแรกของสตริง o_2 เป็น 5 นำมาพิจารณาค่าที่เป็น 5 (ตำแหน่งที่ 6) ของสตริง o_1 จะคงเดิมไว้ ส่วนค่าที่เป็น 6 (ตำแหน่งที่ 6) ของสตริง o_2 จะคงเดิมไว้เช่นกัน จะได้เป็น

o_1 = [2 - - - - 5 - - - -]
o_2 = [5 - - - - 6 - - - -]

จากค่าที่เป็น 6 ของสตริง o_2 นำไปพิจารณาต่อในสตริง o_1 ซึ่งจะคงเดิมไว้ ได้สตริงเป็น

o_1 = [2 - - - - 5 - - - 6]
o_2 = [5 - - - - 6 - - - 4]

ด้วยกระบวนการเดิมจะได้สตริงเป็น

o_1 = [2 - - 4 - 5 - - - 6]

o_2 = [5 - - 1 - 6 - - - 4]

↓

o_1 = [2 - - 4 - 5 - 1 - 6]

o_2 = [5 - - 1 - 6 - 3 - 4]

↓

o_1 = [2 - 3 4 - 5 - 1 - 6]

o_2 = [5 - 2 1 - 6 - 3 - 4]

กระบวนการในการเลือกตัวที่จะคงเดิมไว้จะสิ้นสุดต่อเมื่อ ค่าในตำแหน่งใดตำแหน่งหนึ่งของ o_2 มีค่าตรงกับค่าในตำแหน่งแรกของ สตริง o_1 จากนั้นจะทำการสลับค่าที่ยังคงเหลืออยู่ทั้งสองสตริงแบบ ตำแหน่งต่อตำแหน่ง ได้สตริงลูกหลานเป็น

o_1 = [2 9 3 4 10 5 7 1 8 6]

o_2 = [5 10 2 1 8 6 9 3 7 4]

4) วิธี Order Crossover (OX)

เช่นเดียวกับวิธีการอื่นๆ อย่างแรกคือการเลือกคู่สตริงพ่อแม่ขึ้นมาอย่างสุ่ม จากนั้นทำการเลือกตำแหน่งที่จะทำการครอสโอเวอร์อย่างสุ่ม ขอบเขตของการครอสโอเวอร์อยู่ในช่วงเครื่องหมาย " | "

p_1 = [8 9 4 | 5 6 7 | 1 3 2 10]

p_2 = [8 7 1 | 2 3 10 | 9 5 4 6]

สุ่มตำแหน่งที่จะทำการครอสโอเวอร์ได้อยู่ในช่วง [4,6] จากนั้นเลือกคู่แม๊ปที่อยู่ในช่วงการครอสโอเวอร์ถ้าค่าที่ตรงกันให้ทำเครื่องหมาย # เครื่องหมาย # หมายความว่าปล่อยตำแหน่งนั้นให้ว่าง สตริงลูกหลานจะเป็น

o_1 = [8 9 4 | 5 6 7 | 1 # # #]

o_2 = [8 # 1 | 2 3 10 | 9 # 4 #]

จากนั้นจะทำการเลื่อนตำแหน่ง โดยยึดถือตำแหน่งหลังช่วง " | " เป็นจุดอ้างอิง (สำหรับสตริง o_1 เป็น 1 และ สำหรับ o_2 เป็น 9) ได้สตริงลูกหลานดังต่อไปนี้

o_1 = [5 6 7 | # # # | 1 8 9 4]

o_2 = [2 3 10 | # # # | 9 4 8 1]

ขั้นตอนต่อไปจะเป็นการสลับตำแหน่งที่อยู่ในช่วงของการครอสโอเวอร์ของสตริงทั้งสองที่เหลือ สตริงที่ได้เมื่อทำการครอสโอเวอร์แล้วคือ

o_1 = [5 6 7 | 2 3 10 | 1 8 9 4]

o_2 = [2 3 10 | 5 6 7 | 9 4 8 1]

5) วิธี Position-Base Method (PBX)

วิธีการนี้ถูกพัฒนาขึ้นโดย Syswerda โดยดัดแปลงมาจากวิธีการครอสโอเวอร์แบบ OX (Gen, M. and Chen, R., 1997) โดยเริ่มจากการเลือกสตริงพ่อแม่

p_1 = [2 10 3 4 8 5 9 1 7 6]

p_2 = [5 9 2 1 10 6 7 3 8 4]

จากนั้นเลือกตำแหน่งจากสตริงพ่อแม่ p_1 อย่างสุ่ม แล้วนำค่าในตำแหน่งที่เลือกของสตริงพ่อแม่ p_1 ไปใส่ในตำแหน่งเดียวกันของสตริงลูก o_1 ในที่นี้เลือกตำแหน่งที่ 4 และ 7

p_1 = [2 10 3 4 8 5 9 1 7 6]

p_2 = [5 9 2 1 10 6 7 3 8 4]

สตริงลูกที่ได้จากตำแหน่งของสตริงพ่อแม่ p_1 แบบสุ่ม คือ สตริง o_1

o_1 = [# # # 4 # # 9 # # # #]

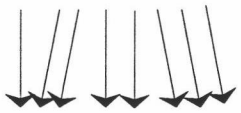
ตัดค่าที่อยู่ในตำแหน่งที่เลือกของสตริงพ่อแม่ p_1 ออกจากสตริงพ่อแม่ p_2 ดังนั้นค่าที่ตัดออกคือ 4 และ 9

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

p_2 = [5 # 2 1 10 6 7 3 8 #]

นำค่าที่เหลืออยู่ในสตริงพ่อแม่ p_2 มาใส่ในสตริงลูก o_1 ตามลำดับ

p_2 = [5 # 2 1 10 6 7 3 8 #]



o_1 = [5 2 1 4 10 6 9 7 3 8]

6.2.7 การมิวเตชัน (Mutation)

การมิวเตชันคือการสลับตำแหน่งของค่าภายในสตริงคำตอบตัวเดียว วิธีการมิวเตชันมี 4 วิธี ซึ่งการพิจารณาว่าสตริงตัวใดจะถูกนำมามิวเตชันขึ้นอยู่กับค่าความน่าจะเป็นในการมิวเตชัน (Pm) โดยการพิจารณาจะเริ่มจากการสุ่มค่า r ซึ่งมีค่าระหว่าง [0,1] ให้กับสตริงคำตอบทุกตัวใน Mating Pool จากนั้น ทำการเลือกเฉพาะสตริงที่มีค่า r น้อยกว่าค่าความน่าจะเป็นในการมิวเตชัน (Pm) ไปทำการมิวเตชัน รายละเอียดในการมิวเตชันของแต่ละวิธีมีดังนี้

1) วิธี Inversion

วิธี Inversion เป็นวิธีการมิวเตชันโดยการสลับค่าของสตริงย่อยภายในสตริงคำตอบ โดยเริ่มต้นจากการสุ่มตัวเลขสุ่มขึ้นมา 2 ตัว จากสตริงที่ต้องการทำการมิวเตชัน p_1 ตัวเลขสุ่มที่มีค่าน้อยกว่าจะเป็นตำแหน่งเริ่มต้นของสตริงคำตอบย่อย ส่วนตัวเลขสุ่มที่มีค่ามากจะเป็นตำแหน่งสุดท้ายภายในสตริงคำตอบย่อย

p_1 = [1 2 3 4 5 6 7 8 9 10]

สมมติว่าทำการสุ่มตัวเลขขึ้นมา 2 ตัวคือ 3 และ 6 สตริงย่อยที่ใช้ทำการสลับตำแหน่งคือ [3 4 5 6] หรือ

p_1 = [1 2 | 3 4 5 6 | 7 8 9 10]

โดยให้ตัวเลขที่อยู่ระหว่าง | แทนสตริงคำตอบย่อย จากนั้นนำสตริงคำตอบย่อยที่ได้มาทำการสลับตำแหน่งโดยให้เรียงลำดับจากขวามาซ้าย ซึ่งตัว

เลขตัวสุดท้ายจะกลายเป็นตัวเลขตัวแรก ส่วนตัวเลขตัวแรกก็จะกลายเป็นตัวเลขตัวสุดท้ายของสตริงคำตอบย่อย

p_1 = [1 2 | 3 4 5 6 | 7 8 9 10]

↓

p_1 = [1 2 | 6 5 4 3 | 7 8 9 10]

2) วิธี Insertion

วิธี Insertion เป็นวิธีการย้ายตำแหน่งของตัวเลขภายในสตริงคำตอบ โดยเริ่มจากการสุ่มตัวเลขของตำแหน่งที่ทำการย้ายคำตอบ ในสตริง p_1 ซึ่งสมมติให้สุ่มได้เลข 6 (เลขที่ขีดเส้นใต้)

p_1 = [1 2 3 4 5 6 7 8 9 10]

จากนั้นทำการสุ่มตัวเลขอีกตัวเพื่อหาตำแหน่งที่จะทำไปแทรกภายในสตริงคำตอบเดียวกัน ซึ่งสมมติให้สุ่มได้เลข 3 (เลขที่ขีดเส้นใต้) เลขที่ได้รับการคัดเลือก (6) ก็จะไปอยู่แทนที่ในตำแหน่งที่ทำการสุ่ม (3)

p_1 = [1 2 3 4 5 6 7 8 9 10]

↓

p_1 = [1 2 6 3 4 5 7 8 9 10]

3) วิธี Displacement

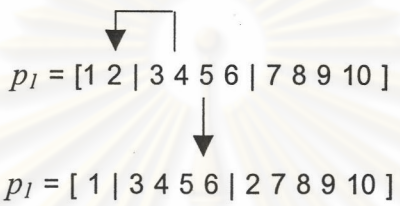
วิธี Displacement เป็นการย้ายตำแหน่งซึ่งคล้ายคลึงกับ วิธี Insertion แต่จะทำการย้ายตำแหน่งของสตริงย่อย ในขั้นแรกจะทำการคัดเลือกสตริงย่อยมาก่อน ซึ่งมีวิธีการคัดเลือกเช่นเดียวกับ วิธี Inversion คือ สุ่มตัวเลขสุ่มขึ้นมา 2 ตัว จากสตริง p_1 ตัวเลขสุ่มที่มีค่าน้อยกว่าจะเป็นตำแหน่งเริ่มต้นของสตริงคำตอบย่อย ส่วนตัวเลขสุ่มที่มีค่ามากจะเป็นตำแหน่งสุดท้ายภายในสตริงคำตอบย่อย

p_1 = [1 2 3 4 5 6 7 8 9 10]

สมมติว่าทำการสุ่มตัวเลขขึ้นมา 2 ตัวคือ 3 และ 6 สตริงย่อยที่ใช้ทำการสลับตำแหน่งคือ [3 4 5 6] หรือ

p_1 = [1 2 | 3 4 5 6 | 7 8 9 10]

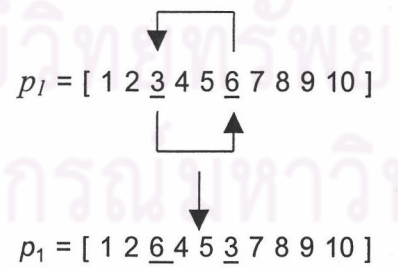
โดยให้ตัวเลขที่อยู่ระหว่าง | แทนสตริงคำตอบย่อย จากนั้นสุ่มตัวเลขขึ้นมาอีกตัวหนึ่งเพื่อเลือกตำแหน่งที่จะทำการแทรกสตริงย่อยที่ได้รับการคัดเลือก สมมติว่าได้เลข 2 จากนั้นทำการแทรกสตริงย่อยนี้ลงไปยังตำแหน่งที่ทำการสุ่ม



จะเห็นได้ว่าหากเราทำการสุ่มตัวเลขของสตริงคำตอบย่อยได้ตัวเลขเดียวกัน จะได้ว่าสตริงคำตอบย่อยนั้นเป็นตำแหน่งเดียวกัน ซึ่งการทำมิวเตชั่นจะเป็นการมิวเตชั่นแบบวิธี Insertion (ย้ายตำแหน่งตัวเลขตัวเดียวภายในสตริงคำตอบ)

4) วิธี Reciprocal Exchange

วิธี Reciprocal Exchange เป็นการสลับตำแหน่งของตัวเลข 2 ตัวภายในสตริงคำตอบ p_1 โดยเริ่มจากทำการสุ่มตัว 2 ตัวที่ไม่ซ้ำกัน สมมติให้สุ่มได้เลข 3 และ 6 จากนั้นทำการสลับตำแหน่งของตัวเลขทั้งสอง



6.2.8 เทคนิคการเก็บค่าที่ดีที่สุด (Elite Preserve Strategy)

เนื่องจากสตริงคำตอบที่ได้จากการครอสโอเวอร์และการมิวเตชัน อาจเป็นคำตอบที่แย่กว่าคำตอบที่เคยปรากฏในเจนเนอเรชันที่ผ่านๆมา ดังนั้นจึงต้องมีการเก็บค่าที่ดีที่สุด (Elite Preserve Solution) เอาไว้เพื่อใช้เปรียบเทียบกับค่าที่ดีที่สุดของสตริงคำตอบชุดใหม่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ถ้าหาก Elite Preserve Solution ให้ค่า Fitness ที่ดีกว่าค่าที่ดีที่สุดของสตริงชุดใหม่ก็ให้เอา Elite Preserve Solution แทนที่ค่าที่แย่ที่สุด ทั้งนี้เพื่อให้สตริงคำตอบที่ดีที่สุดเท่าที่พบยังคงอยู่ในกระบวนการของเจนเนติกอัลกอริทึมต่อไป

เทคนิคการเก็บค่าที่ดีที่สุดจะถูกนำไปใช้ระหว่างกระบวนการของเจนเนติกอัล กอริทึม 3 ครั้ง คือ ดังนั้นจึงแบ่งเทคนิคการเก็บค่าที่ดีที่สุดออกเป็น 3 เทคนิค คือ

1) Initial Elite Preserve Strategy

เป็นจุดเริ่มต้นของเทคนิคการเก็บค่าที่ดีที่สุด เพื่อเก็บค่าที่ดีที่สุดจากคำตอบของสตริงคำตอบในการสร้างประชากรเบื้องต้น ซึ่งกันสตริงคำตอบที่ดีที่สุดหลุดจากประชากรหลังจากทำการคัดเลือกสตริงคำตอบ โดยจะกระทำเพียงครั้งเดียวภายหลังจากการสร้างสตริงคำตอบเบื้องต้น ในตอนต้นของกระบวนการเจนเนติกอัลกอริทึม และผ่านการถอดรหัสรวมทั้งการประเมินค่าเรียบร้อยแล้ว ค่า Fitness ของสตริงแต่ละตัวที่ได้จากการประเมินค่าจะถูกเรียงลำดับจากมากไปน้อย สตริงคำตอบเพียงหนึ่งตัวที่ให้ค่า Fitness มากที่สุดก็จะถูกเลือกไปเป็น คำตอบที่ดีที่สุดที่เก็บไว้ (Elite Preserve Solution) จากนั้นสตริงคำตอบทั้งหมดรวมทั้งตัวที่เลือกไปเป็น Elite Preserve Solution และจะเข้าสู่ขั้นตอนต่างๆ ของเจนเนติกอัลกอริทึม ต่อไป

2) Post-crossover Elite Preserve Strategy

เป็นเทคนิคการเก็บค่าที่ดีที่สุดที่ใช้ภายหลังจากที่เสร็จสิ้นกระบวนการครอสโอเวอร์แล้ว ทั้งนี้เนื่องจากว่าสตริงคำตอบที่ได้จากการครอสโอเวอร์อาจเป็นคำตอบที่ดีกว่าคำตอบอื่นๆที่เคยพบมา แต่เมื่อนำไปมิวเตชันแล้ว สตริงคำตอบตัวนี้จะเปลี่ยนไปและอาจให้คำตอบที่ดีน้อยกว่าเดิม ดังนั้น เพื่อป้องกันไม่ให้สตริงคำตอบที่ดีที่ได้หลังจากการครอสโอเวอร์สูญหายไปจึงต้องทำการการถอดรหัสและประเมินค่าสตริงคำตอบภายหลังการครอสโอเวอร์ทั้งหมด *popsize* ตัว แล้วนำสตริงคำตอบที่ดีที่สุดภายหลังการครอสโอเวอร์ไปเปรียบเทียบกับ Elite Preserve Solution ถ้าหากสตริงคำตอบภายหลังการครอสโอเวอร์ดีกว่า ก็ให้เอาสตริงคำตอบที่ดีที่สุดนั้นไปเป็น Elite Preserve Solution แทน แต่ถ้า Elite Preserve Solution ดีกว่า ก็ให้นำสตริงคำตอบภายหลังจากการครอสโอเวอร์ทั้งหมด *popsize* ตัวไปผ่านกระบวนการมิวเตชันตามปกติ

ยกตัวอย่างเช่น ภายหลังการครอสโอเวอร์ มีสตริงคำตอบ 5 ตัว ซึ่งเมื่อนำไปผ่านกระบวนการถอดรหัสและประเมินค่าแล้วได้ค่า Fitness ของสตริงคำตอบแต่ละตัวเป็น 934 933 925 941 และ 943 ค่า Fitness ที่ดีที่สุดใน 5 ตัวนี้ คือค่า 943 ของสตริงคำตอบตัวที่ 5 ก็ให้เอาค่าวัตถุประสงค์ (ในงานวิจัยนี้จะใช้ค่าเฉลี่ยของเวลาการปรับตั้งรวมและค่าพิสัยของเวลาในการปรับตั้ง เป็นค่าวัตถุประสงค์) ของสตริงคำตอบตัวที่ 5 นี้ไปเปรียบเทียบกับค่าวัตถุประสงค์ของ Elite Preserve Solution ถ้าหากค่าวัตถุประสงค์ของ Elite Preserve Solution มีค่าที่ดีกว่าค่าวัตถุประสงค์ของสตริงคำตอบที่ดีที่สุดจากการครอสโอเวอร์ ก็ให้เอาสตริงคำตอบตัวที่ 5 นี้ไปใช้เป็น Elite Preserve Solution ตัวใหม่แทน แต่ถ้าค่าวัตถุประสงค์ของ Elite Preserve Solution ไม่ดีกว่าหรือเท่ากับค่าวัตถุประสงค์ของสตริง คำตอบที่ดีที่สุดจากการครอสโอเวอร์ ก็ให้คงคำตอบของ Elite Preserve Solution ไว้ แล้วนำสตริงคำตอบทั้ง 5 ตัวนี้ไปทำการมิวเตชันต่อไป

3) Elite Preserve Strategy of Generation

เป็นเทคนิคการเก็บค่าที่ดีที่สุดที่ใช้ภายหลังการมิวเตชัน ซึ่งถือว่าเป็นการเก็บค่าที่ดีที่สุดของเจนเนอเรชันนั้นๆด้วย การเก็บค่าที่ดีที่สุดของเจนเนอเรชันจะช่วยให้คำตอบที่ดีที่สุดเท่าที่เคยปรากฏขึ้นมายังคงมีอยู่ในเจนเนอเรชันต่อไป การเก็บค่าในขั้นตอนนี้จะทำหลังจากที่มีการมิวเตชันเรียบร้อยแล้ว สตริงคำตอบที่ได้ภายหลังการมิวเตชันจำนวน *popsize* ตัว จะถูกถอดรหัสและประเมินค่า จากนั้นก็ให้เอาสตริงคำตอบหลังที่ดีที่สุดจากการมิวเตชัน มาเปรียบเทียบกับ Elite Preserve Solution เช่นเดียวกับในขั้นตอนของ Post-crossver Elite Preserve Strategy แต่แตกต่างกันตรงที่ จะมีการนำเอา Elite Preserve Solution มาแทนที่คำตอบที่แย่ที่สุดของสตริงคำตอบชุดนี้เมื่อ Elite Preserve Solution เป็นคำตอบที่ดีกว่า

ยกตัวอย่างเช่น ภายหลังการมิวเตชันได้สตริงคำตอบ 5 ตัว ที่มีค่า Fitness เป็น 934 933 925 941 และ 943 จะได้ว่าค่า Fitness ที่ดีที่สุดคือ 943 ของสตริงคำตอบตัวที่ 5 ซึ่งถ้าค่าวัตถุประสงค์ของสตริงคำตอบตัวที่ 5 มีค่าที่ดีกว่าค่าวัตถุประสงค์ของ Elite Preserve Solution สตริงคำตอบตัวที่ 5 นี้ จะกลายเป็น Elite Preserve Solution ตัวใหม่ แต่ถ้าค่าวัตถุประสงค์ของ Elite Preserve Solution ดีกว่า ก็ให้ตัดสตริงคำตอบตัวที่ 3 ซึ่งมีค่า Fitness ต่ำที่สุดทิ้งไป เอาสตริงคำตอบที่เป็น Elite Preserve Solution ขณะนั้นไปใส่แทน

สตริงคำตอบที่ได้ภายหลังจากขั้นตอนนี้จะกลายเป็นสตริงคำตอบพ่อแม่ที่แท้จริงในเจนเนอเรชันต่อไป และเมื่อถึงขั้นตอนของการหยุดของกระบวนการเจนเนติกอัลกอริทึม คำตอบสุดท้ายที่เป็น Elite Preserve Solution จะเป็นคำตอบที่ดีที่สุดที่ได้จากกระบวนเจนเนติกอัลกอริทึม

ในการเก็บค่าที่ดีสุดของสตริงที่ผ่านกระบวนการเจนเนติกอัลกอริทึมในแต่ละรอบ (Generation) จะทำให้สตริงที่เป็นคำตอบที่ดีที่สุดไม่เกิดการสูญหายจากวิธีการต่างๆ ของเจนเนติกอัลกอริทึม เช่น การคัดเลือกสตริงคำตอบ การครอสโอเวอร์ และการมิวเตชัน นอกจากนี้ยังเป็นการทำให้ค่าวัตถุประสงค์ที่ได้จากคำตอบในการทำเจนเนติกอัลกอริทึมแต่ละรอบ ไม่มีค่าวัตถุประสงค์ที่ลดลง เนื่องจากเทคนิคในการเก็บค่าที่ดีที่สุดนี้ จะเก็บค่าเฉพาะคำตอบที่ดีกว่าคำตอบเดิมที่ได้จากเจนเนติกอัลกอริทึม ซึ่งเป็นคุณสมบัติอีกประการหนึ่งของเจนเนติกอัลกอริทึมในการพัฒนาคำตอบ

6.3 สรุปท้ายบท

วิธีการเจนเนติกอัลกอริทึมที่ใช้สำหรับปัญหาการจัดตารางการผลิตที่มีเวลาปรับตั้งเครื่องแบบฟัซซี่ซึ่งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้า แบ่งเป็น 5 ส่วนหลักคือการสร้างประชากรเบื้องต้น (Initialization) การรีโปรดักชัน (Reproduction) การครอสโอเวอร์ (Crossover) การมิวเตชัน (Mutation) และเทคนิคการเก็บค่าที่ดีที่สุด (Elite Preserve Strategy)

การสร้างประชากรเบื้องต้นทำได้โดยการสร้างคำตอบเบื้องต้นจำนวนหนึ่งในรูปของสตริงคำตอบที่ได้จากการใส่รหัสแบบ Non-binary string ในส่วนของการรีโปรดักชันจะแบ่งเป็น 3 ส่วนย่อยคือการถอดรหัส (Decoding) ซึ่งเป็นการนำผลิตภัณฑ์ต่างๆในสตริงคำตอบมาจัดตารางการผลิตให้กับเครื่องจักรตามลำดับ การประเมินค่า (Evaluation) เป็นการคำนวณหาค่าความเหมาะสมของสตริงคำตอบแต่ละตัว และการคัดเลือก (Selection) ซึ่งเป็นการคัดเลือกสตริงตัวที่มีความเหมาะสมมากกว่าเพื่อเข้าสู่กระบวนการถัดไปโดยมีทั้งวิธี Roulette Wheel และ Tournament เมื่อเข้าสู่กระบวนการ ครอสโอเวอร์ สตริงจะถูกสุ่มเลือกด้วยความน่าจะเป็นในการครอสโอเวอร์เพื่อมาจับคู่และทำการครอสโอเวอร์ วิธีครอสโอเวอร์ที่ใช้มี 5 วิธีคือ MOX, PMX, CX, OX และ PBX จากนั้นสตริงคำตอบจะถูกสุ่มเลือกไปทำการมิวเตชันด้วยความน่าจะเป็นในการมิวเตชัน วิธีมิวเตชันที่ใช้อย่างแพร่หลายคือวิธี Inversion, Insertion, Displacement และ Reciprocal Exchange สตริงคำตอบที่ได้จะกลายเป็นประชากรพ่อแม่ในเจนเนอเรชันต่อไป กระบวนการของเจนเนติกอัลกอริทึม จะเกิดขึ้นซ้ำไปเรื่อยๆจนกว่าจะถึงเจนเนอเรชันสูงสุดที่กำหนด

ในระหว่างกระบวนการของเจนเนติกอัลกอริทึม มีการนำเทคนิคการเก็บค่าที่ดีที่สุดเข้าไปใช้ภายหลังการคัดเลือกสตริงคำตอบเบื้องต้น การครอสโอเวอร์ และการมิวเตชัน ทั้งนี้เพื่อให้สตริงคำตอบที่ดียังคงอยู่ต่อไปในเจนเนอเรชันถัดไป และเมื่อถึงเงื่อนไขของการหยุดของกระบวนการของเจนเนติกอัลกอริทึมคำตอบสุดท้ายที่ได้จากการเก็บค่าที่ดีที่สุด จะเป็นคำตอบที่ดีที่สุดที่ได้จากกระบวนการของเจนเนติกอัลกอริทึม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย