

บทที่ 2

ทฤษฎีที่เกี่ยวกับจีเนติกอัลกอริทึม

เนื้อหาในบทนี้กล่าวถึงทฤษฎีเบื้องต้นของจีเนติกอัลกอริทึม (Genetic Algorithm: GAs) และตัวอย่างการคำนวณหาค่าของคำตอบของจีเนติกอัลกอริทึมอย่างง่าย เพื่อเป็นพื้นฐานสำหรับการนำจีเนติกอัลกอริทึมไปใช้ในการแก้ปัญหาคำถาม Optimization แบบต่างๆ

2.1 จีเนติกอัลกอริทึม

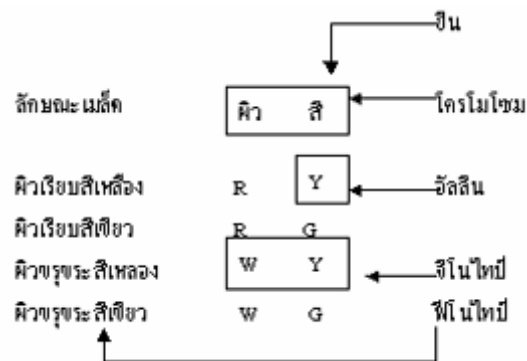
ในปัจจุบันการหาคำตอบของปัญหาบางประเภท เช่น ปัญหาการจัดสรรทรัพยากรที่มีอยู่อย่างจำกัดเพื่อให้เกิดประสิทธิภาพสูงสุดในการแก้ปัญหาในการคำนวณต้นทุนที่ต่ำสุด เป็นต้น สามารถหาคำตอบได้หลายวิธี วิธีที่ง่ายที่สุดในการหาคำตอบคือวิธีการทางฮิวริสติกต่างๆ ซึ่งอาจได้คำตอบที่ไม่ดีนัก ในปัจจุบันนักวิทยาศาสตร์ได้นำความรู้เกี่ยวกับทฤษฎีหรือกฎเกณฑ์ทางธรรมชาติมาช่วยในการหาคำตอบหรือการศึกษาวิจัย โดยมีเป้าหมายในการใช้ประโยชน์ของความคงทน (Robustness) ความไม่เที่ยงตรงแม่นยำ (Accuracy) ความไม่แน่นอน (Uncertainty) หรือความคลุมเครือของปัญหา (Vague) หลักการเหล่านี้สามารถพบได้จากวิธีการต่างๆ เช่น ระบบโครงข่ายประสาทเทียม (Neural Network) ฟัซซี่ลอจิก (Fuzzy Logic) และ Gas (Coldberg, 1989)

ปัญหาที่พบส่วนใหญ่เป็นปัญหาที่ไม่เที่ยงตรงและมีความคลุมเครือ ซึ่งถ้าหากต้องการคำตอบที่เที่ยงตรงและมีความแน่นอนสูงมากก็ย่อมมีค่าใช้จ่ายที่สูงมาก ดังนั้นวิธีที่สามารถแก้ปัญหาที่คลุมเครือโดยได้คำตอบที่ใกล้เคียงสามารถยอมรับได้ ใช้เวลาในการหาคำตอบไม่มากนัก และมีค่าใช้จ่ายพอประมาณ ย่อมดีกว่าวิธีได้ความเที่ยงตรงสูงแต่มีค่าใช้จ่ายสูง วิธีการหาคำตอบที่ได้อย่างหนึ่งได้แก่วิธีการของ GAs โดยเอาทฤษฎีในการถ่ายทอดลักษณะต่างๆทางกรรมพันธุ์ไปยังลูกหลาน ซึ่งสามารถนำมาพัฒนาใช้ในการหาคำตอบที่ต้องการได้

2.1.1 พันธุศาสตร์กับจีเนติกอัลกอริทึม

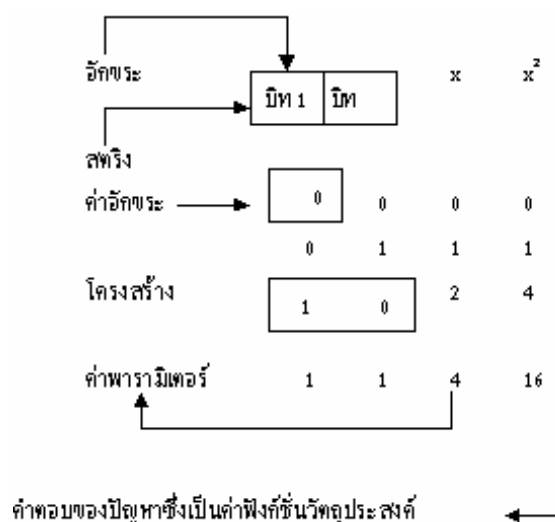
Mendel บิดาแห่งวิชาพันธุศาสตร์ ค้นพบลักษณะต่างๆของสิ่งมีชีวิต เช่น ลักษณะของผิวเมล็ดพืชสีของเมล็ดพืช ฯลฯ ที่ถูกถ่ายทอดไปยังลูกหลานนั้นถูกควบคุมโดยหน่วยควบคุมลักษณะที่เรียกว่า ยีน (Gene) และลักษณะย่อยของยีนเรียกว่าอัลลีล (Allele) เช่น ยีนควบคุมลักษณะของผิวเมล็ดพืชจะมีอัลลีลเป็นผิวเรียบและผิวขรุขระ เป็นต้น ซึ่งแต่ละยีนจะเรียงตัวอยู่บนโครโมโซม (Chromosome) ภายในเซลล์ ตำแหน่งของแต่ละยีนบนโครโมโซมเรียกว่า โลกัส (Locus) และแต่ละแบบของชุดยีนเรียกว่า จีโนไทป์ (genotype) ซึ่งลักษณะภายนอกที่ปรากฏ ซึ่งเรียกว่า ฟิโนไทป์ (Phenotype) ดังรูปที่ 2.1 ซึ่งแสดงให้เห็น

ถึงลักษณะทางพันธุศาสตร์ของโครโมโซมควบคุมลักษณะของเมล็ดถั่ว ซึ่งมียีนลักษณะของผิวเมล็ดคือ มีลักษณะเรียบ (R) หรือขรุขระ(W) และยีนลักษณะสีของเมล็ดคือ มีสีเหลือง(Y)และสีเขียว(G)



รูปที่ 2.1 แสดงลักษณะทางพันธุศาสตร์ของโครโมโซมควบคุมลักษณะของเมล็ดถั่ว

การแก้ปัญหามathematics ด้วย GAs พารามิเตอร์ต่างๆจะถูกแปลงให้อยู่ในรูปของสตริง (string) หรือโครโมโซมประกอบด้วยอักขระ (character) หรือบิต (bit) แต่ละตำแหน่งของโครโมโซมที่ให้คำตอบของปัญหาแตกต่างกัน ดังรูปที่ 2.2 ซึ่งเป็นการประยุกต์ใช้ GAs กับการแก้ปัญหาค่าสูงสุดของฟังก์ชัน $f(x)=x^2$ โดยที่ x อยู่ในช่วง $[0,4]$ และค่าของ x จะถูกแปลงให้อยู่ในรูปไบนารีสตริง ซึ่งสามารถสรุปความหมายทางพันธุศาสตร์เทียบกับ GAs ได้ดังตารางที่ 2.1



รูปที่ 2.2 ลักษณะทางคณิตศาสตร์ของการแก้ปัญหาในการหาค่าสูงสุดของ $f(x) = x^2$

ตารางที่ 2.1 เปรียบเทียบคำศัพท์ระหว่างพันธุศาสตร์กับเจนเนติกอัลกอริทึม

พันธุศาสตร์	เจนเนติกอัลกอริทึม
โครโมโซม (Chromosome)	สตริง (String)
ยีน (Gene)	คุณลักษณะ, สตริง (character} bit)
อัลลีล (Allele)	ค่าของคุณลักษณะ, ค่าบิต (Character value} bit value)
โลคัส (Locus)	ตำแหน่ง (String position)
จีโนไทป์ (Genotype)	โครงสร้าง (Structure)
ฟีโนไทป์ (Phenotype)	โครงสร้างคำตอบ (A decode structure)

Darwin (1859) ได้เสนอความคิดการเกิดสปีชีส์ของสิ่งมีชีวิต (the origin of species) โดยเสนอหลักการของวิวัฒนาการที่ผ่านกระบวนการคัดเลือกตามธรรมชาติแม้ในตอนแรกทฤษฎีจะเป็นที่โต้แย้งกันมาก ต่อมาก็ได้เป็นที่ยอมรับในหมู่นักวิทยาศาสตร์ (Winston, 1992) ไว้ดังนี้

- สิ่งมีชีวิตแต่ละชนิดมีแนวโน้มที่จะถ่ายทอดลักษณะของมันไปสู่ลูกหลานของมัน
- ธรรมชาติทำให้สิ่งมีชีวิตมีลักษณะต่างๆกัน
- สิ่งมีชีวิตมีความเหมาะสม ซึ่งมีลักษณะที่เหมาะสมที่สุด มีแนวโน้มที่จะมีลูกหลานมากกว่าสิ่งมีชีวิตที่มีลักษณะไม่เหมาะสม ซึ่งจะทำให้ประชากรอยู่รอดต่อไป
- เมื่อระยะเวลาผ่านไปยาวนาน จะเกิดการกลายพันธุ์ (variation) ขึ้น และเกิดสปีชีส์ที่มีลักษณะเหมาะสมกับระบบนิเวศนั้น

2.1.2 ความหมายของเจนเนติกอัลกอริทึม

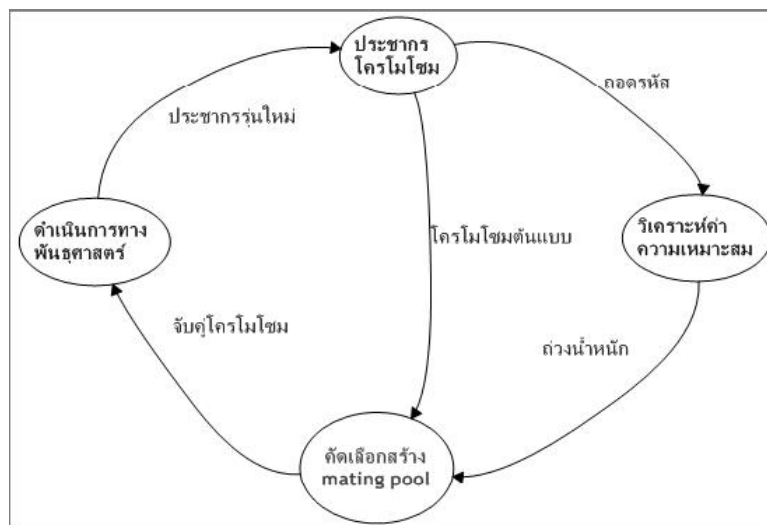
GAs เป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานมาจากกระบวนการคัดเลือกตามธรรมชาติ (Natural Selection) และ กระบวนการคัดเลือกทางพันธุศาสตร์ (Natural Genetics Selection) โดยการคัดเลือกสตริง (String) ที่มีความเหมาะสมของกลุ่มสตริงทั้งหมดด้วยวิธีการสุ่ม จากการนำสตริงเหล่านี้ไปผ่านกระบวนการคัดเลือกสตริงที่มีความเหมาะสม ซึ่งสตริงที่มีความเหมาะสมนี้ คือ คำตอบที่ดีที่สุดหรือใกล้เคียงคำตอบที่ดีที่สุด GAs ไม่ใช่การสุ่มแบบง่ายๆ แต่มันเป็นการใช้ข้อมูลในรูปอดีตอย่างมีประสิทธิภาพ เพื่อพิจารณาจุดที่จะต้องค้นหาใหม่โดยการคาดหวังว่าสมรรถนะของการค้นพบจะดีขึ้น GAs ถูกพัฒนาขึ้นโดย Holland (1975) และคณะ โดยมีเป้าหมายในการวิจัย 2 อย่าง คือ เพื่อสรุปและดัดแปลงการใช้กระบวนการทางธรรมชาติให้ถูกต้องมากที่สุด และเพื่อออกแบบและสร้างซอฟต์แวร์ที่รักษากลไกที่สำคัญของธรรมชาติ ซึ่ง GAs แตกต่างกับวิธีการค้นหาและการทำ Optimization แบบอื่นๆ คือ

- GAsทำงานโดยเข้ารหัสตรงเป็นชุดพารามิเตอร์
- GAsเป็นการค้นหาจากประชากร ไม่ใช่ค้นหาจากเพียงตำแหน่งๆเดียว
- GAsใช้ข่าวสารที่เป็นผลลัพธ์ (ฟังก์ชันเป้าหมาย) โดยไม่ใช่การอนุมานหรือความรู้อื่นๆ
- GAsจะเป็นวิธี probabilistic ไม่ใช่ Deterministic

2.1.3 วงจรของเจนเนติกอัลกอริทึม (Genetic Algorithm Cycle)

การทำงานของเจนเนติกอัลกอริทึมเริ่มต้นด้วย

1. สร้างประชากรต้นกำเนิด (Initial Population) จากการสุ่มสร้างค่าแต่ละบิตของแต่ละโครโมโซม
2. แล้วทำการวิเคราะห์ค่าความเหมาะสมแต่ละโครโมโซม โดยถอดรหัสค่าตัวแปร พารามิเตอร์ต่างๆ ของแต่ละบิตในโครโมโซมและคำนวณค่าความเหมาะสมจากฟังก์ชันความเหมาะสมที่กำหนดไว้
3. หลังจากนั้นทำการสร้างชุดโครโมโซมต้นแบบ (Mating Pool) หรือชุดโครโมโซมพ่อแม่ที่สามารถอยู่รอดเป็นต้นแบบโดยใช้วิธีการคัดเลือกทางธรรมชาติที่พิจารณาจากค่าความเหมาะสมของแต่ละโครโมโซม ถ้าโครโมโซมใดมีค่าความเหมาะสมมากก็จะมีโอกาสถูกคัดเลือกเป็นต้นแบบมาก
4. สุดท้ายก็ดำเนินการทางพันธุศาสตร์โดยการสุ่มจับคู่โครโมโซมต้นแบบเพื่อสร้างประชากรรุ่นใหม่ ซึ่งกระบวนการทำงานดังกล่าวแสดงดังรูปที่ 2.3



รูปที่ 2.3 วงจรของเจนเนติกอัลกอริทึม

2.2 เจนเนติกอัลกอริทึมอย่างง่าย (Simple Genetic Algorithms)



รูปที่ 2.4 ขั้นตอนการทำ GAs อย่างง่าย (Simple Genetic Algorithms:SGA)

ขั้นตอนการทำ GAs อย่างง่าย ดังรูปที่ 2.4 มีรายละเอียดดังต่อไปนี้

2.2.1 การเข้ารหัสและสร้างประชากรเริ่มต้นอย่างสุ่ม

ขั้นแรกของ GAs คือ การเข้ารหัสหรือแปลงค่าพารามิเตอร์ให้อยู่ในรูปของสตริงที่มีความยาวแน่นอน ซึ่งวิธีการเข้ารหัสนี้ขึ้นอยู่กับรูปแบบของปัญหาแต่ละปัญหาสำหรับ SGA ในการเข้ารหัสแบบไบนารี (Binary Coding)

ตัวอย่างเช่น ต้องการหาค่าสูงสุดของฟังก์ชัน $f(x) = x^2$ โดยที่ x มีค่าอยู่ระหว่าง $[0,31]$ ในที่นี้ฟังก์ชันวัตถุประสงค์ (Objective Function) คือ $f(x)$ หรือ x^2 ซึ่งวิธีการเข้ารหัสแบบไบนารี โดยแปลงค่าพารามิเตอร์ x ให้อยู่ในรูปไบนารี 5 บิต จะได้ค่าพารามิเตอร์ของ x ที่มีค่าอยู่ในช่วง 00000 จนถึง 11111 (0 ถึง 31)

เมื่อกำหนด วิธีการเข้ารหัสแล้ว จำเป็นต้องสร้างประชากรเริ่มต้น (Initial Population) โดยวิธีการสุ่มตัวอย่าง เพื่อจะได้ผ่านขั้นตอนของ SGA ต่อไป สมมติว่าสุ่มประชากรเริ่มต้น 4 สตริงจะได้เป็น

01101

11000

01000

10011

ค่าสตริงของประชากรเริ่มต้นนี้เกิดจากการสุ่มค่าทั้งหมด 20 ครั้งคือ สตริงแต่ละตัวทำหารสุ่ม 5 ครั้ง

2.2.2 ประชากรรุ่นเก่า (Old Population)

ประชากรรุ่นเก่า คือสตริงที่จะถูกคัดเลือกไปเป็นต้นแบบสำหรับสร้างประชากรรุ่นใหม่ (New Popoulation) โดยประชากรรุ่นเก่าชุดแรกคือประชากรเริ่มต้นนั่นเอง

2.2.3 การดำเนินการของ SGA

SGA ประกอบไปด้วยตัวปฏิบัติการ 3 อย่างได้แก่ รีโพรดักชัน (Reproduction) การครอสโอเวอร์ (Crossover) และการมิวเตชัน (Mutation) ดังมีรายละเอียดดังต่อไปนี้

- **รีโพรดักชัน (Reproduction)** คือกระบวนการที่สร้างสารแต่ละตัวเลียนแบบค่าฟังก์ชันเป้าหมาย $f(x)$ โดยที่ฟังก์ชันนี้อาจเป็นการวัดผลตอบแทน ค่าอัตราประโยชน์ หรือสิ่งที่ต้องการให้เป็นค่าสูงสุด หรือค่าความเหมาะสม สตริงที่มีความเหมาะสมสูงกว่าก็มีค่าความน่าจะเป็นในการสนับสนุนรุ่นต่อไปได้สูงด้วย ตัวปฏิบัติการนี้เกิดขึ้นจากกระบวนการคัดเลือกตามธรรมชาติ ตามทฤษฎีผู้รอดที่มีความเหมาะสม (Survival of Fittest) ของชาลส์ ดาวิน ประชากรที่มีความเหมาะสมในธรรมชาติจะมีความสามารถในการรอดพ้นผู้ล่า โรคภัยไข้เจ็บ อุปสรรคอื่นๆ ที่ต่อต้านการเจริญเติบโตเป็นผู้ใหญ่จะสามารถสืบพันธุ์ต่อไปได้ ส่วนฟังก์ชันเป้าหมายจะเป็นสิ่งที่ใช้พิจารณาว่าสตริงที่สร้างขึ้นจะมีชีวิตอยู่หรือตายจากไป ซึ่งวิธีการเลือกสรรมีหลากหลายวิธี ดังนี้

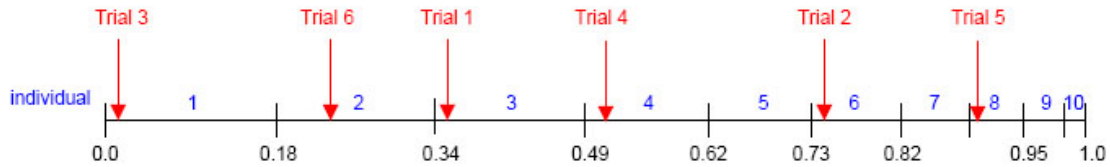
1) Roulette-wheel selection เป็นกระบวนการเลือกสรรแบบ Stochastic sampling ซึ่งมีกระบวนการดังนี้

1.1) นำประชากรแต่ละตัวมาเรียงต่อกันเป็นเส้นตรงความยาวตั้งแต่ 0 ถึง 1 โดยให้ช่วงความยาวของเส้นตรงของแต่ละประชากรขึ้นกับค่าความแข็งแกร่ง คือ หากมีความแข็งแกร่งมาก ก็จะมีช่วงความยาวบนเส้นตรงที่สร้างขึ้นแทนค่าของประชากรตัวนั้นยาว

1.2) สุ่มตัวเลขขึ้นมาเพื่อใช้เป็นตัวชี้ไปยังตำแหน่งบนเส้นตรง

1.3) หากตัวชี้ชี้ไปยังประชากรตัวใดแล้วก็ถือว่าประชากรตัวนั้นเป็นผู้ที่ถูกเลือกเพื่อใช้ในการให้กำเนิดรุ่นถัดไป

1.4) ทำซ้ำตั้งแต่ขั้นตอนที่ 1.1) ถึง 1.3) จนกระทั่งได้ประชากรครบจำนวนที่จะใช้ในการให้กำเนิดรุ่นถัดไป



รูปที่ 2.5 แสดงขั้นตอนการเลือกสรรประชากรแบบ Roulette-wheel selection จากการสุ่ม 6 ครั้ง

2) Stochastic universal sampling เป็นการเลือกสรรโดยจะดำเนินการคล้ายๆ กับวิธี Roulette-Wheel Selection แตกต่างกันเพียงในขั้นตอนสร้างตัวชี้ ซึ่งในวิธีนี้จะให้ระยะห่างระหว่างตัวชี้คงที่ซึ่งสามารถแสดงเป็นขั้นตอนได้ดังนี้

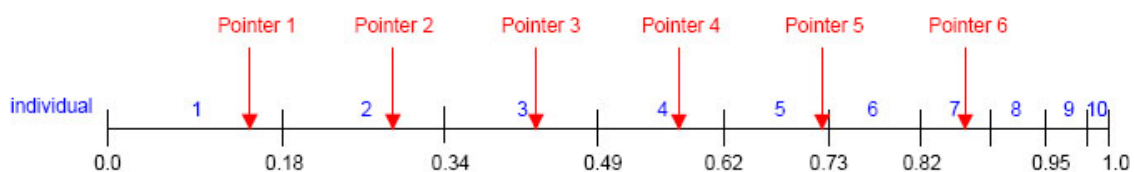
2.1) นำประชากรแต่ละตัวมาเรียงต่อกันเป็นเส้นตรงความยาวตั้งแต่ 0 ถึง 1 โดยให้ช่วงความยาวของเส้นตรงของแต่ละประชากรขึ้นกับค่าความแข็งแรง

2.2) สร้างตัวชี้ โดยจำนวนตัวชี้จะเท่ากับจำนวนประชากรที่ต้องการใช้ในการให้กำเนิดรุ่นถัดไป และตัวชี้แต่ละตัวจะชี้ไปยังตำแหน่ง $1/N_{\text{point}} \times \text{order}$

เมื่อ N_{point} คือ จำนวนตัวชี้ ซึ่งจะเท่ากับจำนวนรุ่นพ่อแม่ที่จะใช้ในการให้กำเนิดรุ่นถัดไป

order คือ ลำดับตัวชี้ มีค่าตั้งแต่ 1 ถึง N_{point}

2.3) หากตัวชี้ชี้ไปยังประชากรตัวใดแล้วก็ถือว่าประชากรตัวนั้นเป็นผู้ที่ถูกเลือกเพื่อใช้ในการให้กำเนิดรุ่นถัดไป



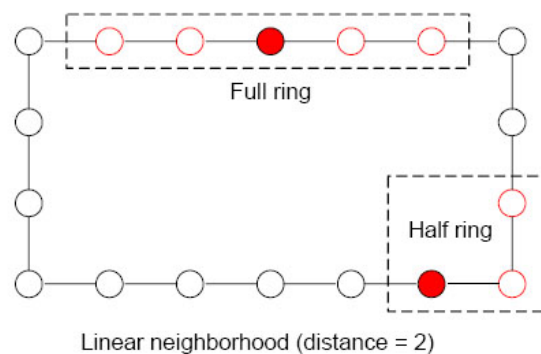
รูปที่ 2.6 ขั้นตอนการเลือกสรรประชากรแบบ Stochastic universal sampling จากจำนวนตัวชี้ 6 ตัว

3) Local selection เป็นวิธีการเลือกสรรที่จะสนใจเฉพาะประชากรที่อยู่ในย่าน (Neighborhood) ที่หนด ซึ่งแตกต่างจาก 2 วิธีแรก (2 วิธีแรก ย่านที่กำหนดจะเป็นทุกประชากรในกลุ่ม) โดยย่านที่กำหนดนี้จะขึ้นอยู่กับโครงสร้างของประชากรแต่ละตัวที่เรียงต่อกันอยู่ และระยะห่าง (Distance) ที่กำหนดหรืออาจกล่าวได้ว่า ย่าน คือ กลุ่มย่อยของประชากรที่เราสนใจเลือก ซึ่งเหมาะกับการวิเคราะห์ปัญหาที่ต้องใช้จำนวนประชากรมากและสามารถแสดงเป็นขั้นตอนได้ดังนี้

- 3.1) เลือกประชากรที่อยู่ในกลุ่ม อาจใช้วิธีสุ่มโดยตรงหรือใช้วิธีการเลือกแบบRoulette-wheel selection หรือ Stochastic universal sampling ก็ได้
- 3.2) กำหนดย่านของประชากรที่ถูกเลือกนี้ พร้อมทั้งกำหนดระยะห่างที่พิจารณา
- 3.3) ในย่านที่กำหนด เลือกประชากรตัวที่มีความแข็งแกร่งที่สุด เพื่อนำมาใช้ในการให้กำเนิดประชากรรุ่นถัดไป
- 3.4) ทำซ้ำตั้งแต่ขั้นตอนที่ 3.1) ถึง 3.3) จนกระทั่งได้ประชากรครบจำนวนที่จะใช้ในการกำหนดรุ่นถัดไป

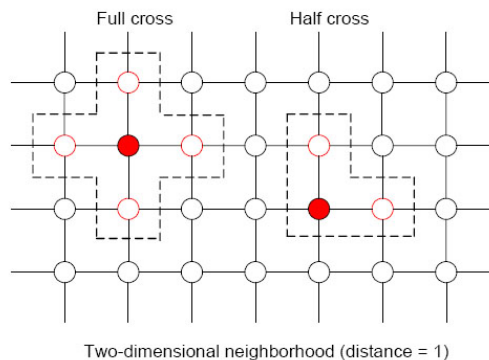
รูปแบบของย่านที่กำหนดจะมีรูปแบบดังนี้

1. Linear จะแบ่งออกเป็น Full ring และ Half ring

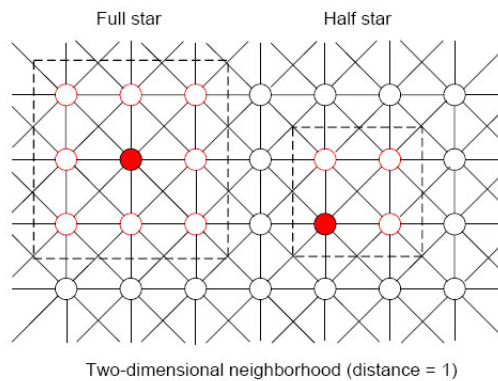


รูปที่ 2.7 แสดงย่านในแบบ Linear ลักษณะ Full ring และ Half ring

2. Two-dimensional จะแบ่งออกเป็น Full cross, Half cross, Full star และ Half star



รูปที่ 2.8 แสดง ย่านในแบบ Two-dimensional ลักษณะ Full cross และ Half cross



รูปที่ 2.9 แสดง ย่านในแบบ Two-dimensional ลักษณะ Full star และ Half star

3. Three-dimensional และวิธีอื่นๆที่มีความซับซ้อนกว่า 2 แบบข้างต้น โดยระยะห่าง(Distance) จะเป็นค่าที่กำหนดจำนวนประชากรในย่าน

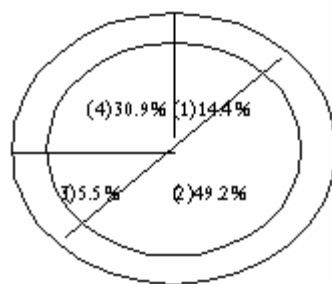
ตารางที่ 2.2 แสดงความสัมพันธ์ระหว่างโครงสร้างและระยะห่าง

Structure	Distance	
	1	2
Full ring	2	4
Half ring	1	2
Full cross	4	8(12)
Half cross	2	4(5)
Full star	8	24
Half star	3	8

4) Truncation selection เป็นวิธีที่คัดเลือกประชากรที่จะใช้ในการให้กำเนิดรุ่นลูก โดยการตัดประชากรที่มีความแข็งแรงน้อยๆ ออกไปก่อน ซึ่งประชากรที่มีความแข็งแรงเกินกว่าค่าขีดจำกัดที่กำหนด (Threshold) เท่านั้นจึงจะมีสิทธิในการได้รับเลือกให้กำเนิดรุ่นลูกต่อไป

5) Tournament selection จะเป็นการคัดสรรโดยจัดกลุ่มย่อยแบบสุ่มให้กับประชากร และประชากรที่มีความแข็งแรงที่สุดในกลุ่มย่อยก็จะได้เป็นผู้ที่ถูกเลือกให้กำเนิดรุ่นลูกต่อไป

ตัวปฏิบัติการรีโพรดักชันสามารถสร้างขึ้นมาได้หลายวิธี วิธีง่ายวิธีหนึ่งคือ สร้างจากวงล้อรูเล็ตที่มีจำนวน ช่องว่าง เท่ากับจำนวนประชากรสตริง และขนาดของช่องก็เป็นสัดส่วนกับค่าความเหมาะสม ดังรูป



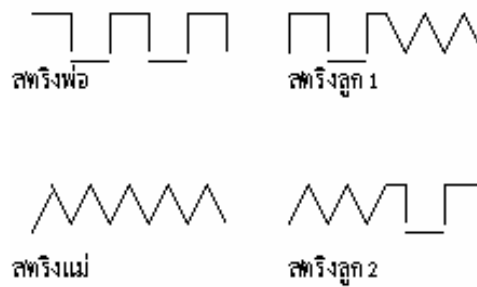
รูปที่ 2.10 แสดงการรีโปรแกรมอย่างง่ายด้วยวิธีการใช้วงล้อสุ่มที่มีขนาดของแต่ละช่องเป็นสัดส่วนกับค่าความเหมาะสม

ตารางที่ 2.3 แสดงกลุ่มประชากรตัวอย่างและค่าความเหมาะสม

No	สตริง	ค่าความเหมาะสม	% โดยรวม
1	0110	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
รวม		1170	100

ค่าความเหมาะสมทั้งหมด โดยรวมจะได้ 1170 และค่ารายละเอียดต่างๆแสดงดังตารางที่ 3.2 ได้แสดงถึงวงล้อสุ่ม สำหรับการรีโปรแกรม ซึ่งสร้างจากสัดส่วนของค่าฟิตเนสของสตริงทั้งหมด เช่น สตริงหมายเลข 1 มีค่าความเหมาะสมเป็น 169 หรือ 14.4% ($169/1170$) ของค่าฟิตเนสโดยรวมของทั้งประชากร ในการทำรีโปรแกรม จะหมุนวงล้อเป็นจำนวน 4 ครั้ง หรือเท่ากับจำนวนสตริง เช่น สตริงหมายเลข 1 มีค่าความเหมาะสมเป็น 169 หรือ 14.4% ของค่าความเหมาะสมทั้งหมด ดังนั้นเมื่อหมุนวงล้อ 1 ครั้ง ก็จะมีแนวโน้มที่จะถูกเลือกเท่ากับ 0.144 ในการหมุนวงล้อแต่ละครั้งจะได้ตัวแทนในการสืบพันธุ์ (Reproduction Candidate) สตริงที่มีความเหมาะสมสูงจะถูกคัดเลือกสำหรับการสืบพันธุ์ การรีโปรแกรมสำหรับสตริงลูกหลานในรุ่นต่อไป เมื่อสตริงมีรูปร่างที่แน่นอนแล้วก็จะถูกส่งไปเข้าเมตติ่งพูล (Mating pool) เพื่อที่จะผ่านกระบวนการของตัวปฏิบัติการอื่นต่อไป

- **การครอสโอเวอร์ (Crossover)** หลังจากประชากรทั้งหมดผ่านกระบวนการรีโปรแกรมแล้ว จะทำการจับคู่สมาชิกในเมตติ่งพูล หรือกลุ่มประชากรทั้งหมดอย่างสุ่มและทำการไขว้สลับค่าที่อยู่หลังตำแหน่งที่เลือกไว้จากการสุ่มหรือ ทำการแลกเปลี่ยนส่วนกัน



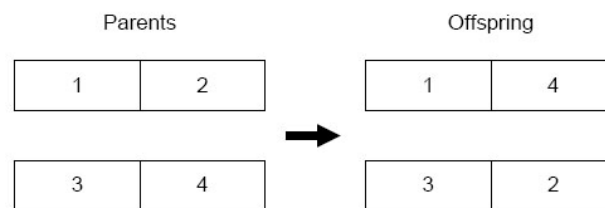
รูปที่ 2.11 แสดงการครอสโอเวอร์อย่างง่ายเพื่อให้เห็นถึงการเปลี่ยนแปลงสตริงและการแลกเปลี่ยนข่าวสารโดยเลือกตำแหน่งไขว้แบบสุ่ม

ขั้นตอนการแลกเปลี่ยนข้อมูลสามารถแบ่งออกได้เป็น 2 กลุ่ม คือ

1. Real valued recombination

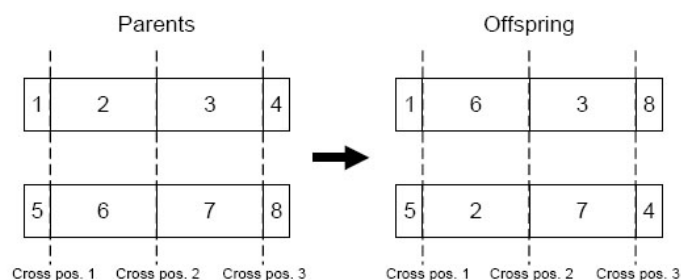
2. Binary valued recombination (Crossover) แบ่งออกเป็น

1) Single-point crossover เป็นกระบวนการที่เริ่มต้นจากการสุ่มตัวเลขที่ใช้แทนตำแหน่งในการตัดต่อข้อมูลจากผู้ให้กำเนิดมายังรุ่นลูก ซึ่งจุดตัดต่อจะเลือกมาเพียงจุดเดียวเท่านั้น



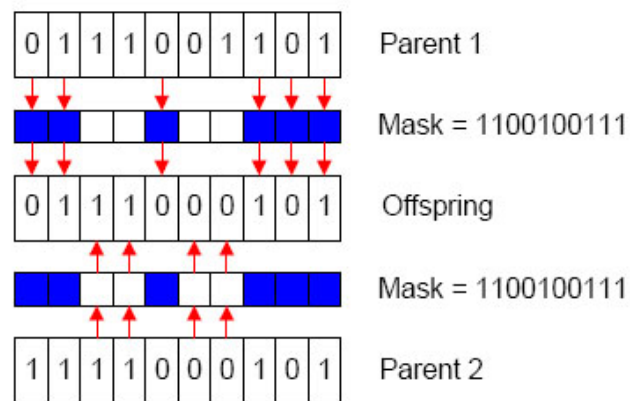
รูปที่ 2.12 แสดง Single-point crossover

2) Multi-point crossover เป็นกระบวนการที่คล้ายกับวิธี Single-point crossover แตกต่างกันที่ ในวิธีนี้จะสุ่มตัวเลขเพื่อใช้ในการตัดต่อข้อมูลจากผู้ให้กำเนิดมายังรุ่นลูกหลายจุด



รูปที่ 2.13 แสดง Multi-point crossover แบบใช้จุดตัด 3 จุด

3) Uniform crossover เป็นวิธีการ Crossover ที่ไม่กำหนดจำนวนจุดตัดที่ใช้ตัดต่อผู้ให้กำเนิด การตัดต่อจะกระทำโดยการสุ่มหน้าต่าง (Mask) จากนั้นอาศัยหน้าต่างที่สุ่มขึ้นในการเลือกแต่ละบิต สำหรับรุ่นลูก



รูปที่ 2.14 แสดง Uniform crossover

4) Shuffle crossover เป็นวิธีที่คล้ายกับการ Crossover ที่กล่าวมาแล้วข้างต้น แตกต่างกันที่ก่อนการนำข้อมูลของผู้ให้กำเนิดมาแลกเปลี่ยนกันเพื่อให้กำเนิดรุ่นลูก จะต้องสลับตำแหน่งบิตก่อนการแลกเปลี่ยนข้อมูล เพื่อลดผลของความสำคัญของลำดับข้อมูล เมื่อแลกเปลี่ยนข้อมูลเสร็จแล้วก็จะจัดลำดับบิตตามเดิม

- **การมิวเตชัน (Mutation)** การมิวเตชันเป็นสิ่งจำเป็นถึงแม้ว่า รีโพรดักชันและครอสโอเวอร์ช่วยให้การค้นหาเป็นไปอย่างมีประสิทธิภาพ บางครั้งก็มีการสูญเสียสิ่งที่สำคัญไป (ค่า 1 หรือ 0 ในบางตำแหน่ง) การมิวเตชันจะป้องกันส่วนที่สูญเสียที่ไม่อาจเรียกคืนได้ () ในบางครั้งการหาคำตอบของSGA คำตอบอาจติดอยู่ใน local optima การมิวเตชันด้วยอัตราส่วนที่เหมาะสมจะทำให้ได้คำตอบที่สามารถหลุดออกจาก local optima หรืออาจกล่าวได้ว่าโอเปอร์เรเตอร์ของการมิวเตชันเป็นการเปลี่ยนแปลงค่าตำแหน่งสตริง จากปัญหาที่พิจารณาค่าจะเปลี่ยนแปลงจาก 0 เป็น 1 หรือ 1 เป็น 0 โดยการลอกตำแหน่งที่จะทำการมิวเตชันอย่างสุ่ม อัตราการมิวเตชันในธรรมชาติจะมีค่อนข้างต่ำ ในการนำไปใช้งานจะต้องมีการพิจารณาอย่างเหมาะสม

2.2.4 ประชากรรุ่นใหม่ (New population)

สตริงทั้งหมดที่ได้จากกระบวนการของ Gasยกว่าประชากรรุ่นใหม่หรือ generation รุ่นใหม่ซึ่งจะกลายเป็นประชากรรุ่นเก่า สำหรับการดำเนินการต่อไป กระบวนการของ SGA จะทำไปเรื่อยๆจนกว่าจำนวน generation จะมากกว่าจำนวน generationที่กำหนดไว้สูงสุด

Surech (1995) ได้พิจารณาถึงการหาขนาดของประชากร จากอัตราส่วนของวิธีการที่ทั้งหมดของคำตอบที่เป็นไปได้และอัตราส่วนของวิธีการทั้งหมดที่เป็นไปได้

$$\lim_{n \rightarrow \infty} \frac{n^2}{n!} \approx \lim_{n \rightarrow \infty} \frac{(2\pi)^{\frac{1}{2}} \left(\frac{n}{e}\right)^n}{n^n} \quad (2.1)$$

$$= \lim_{n \rightarrow \infty} \frac{(2\pi)^{\frac{1}{2}}}{e^n} = 0$$

จากสมการ ให้

ตัวแปร n คือขนาดของปัญหา

ตัวแปร $n!$ คือจำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงที่เป็นไปได้

ตัวแปร n^n คือจำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงทั้งหมด

สามารถสรุปได้ว่า ความน่าจะเป็นในการสร้างประชากรคำตอบอย่างสุ่มมีค่าเป็นศูนย์เมื่อ n มีค่ามากขึ้น สมมติว่า n มีค่าเป็น 8 ดังนั้น $8!/88 = 1/416.1 = 2.403 \times 10^{-3}$ หรืออาจกล่าวได้ว่า โอกาสที่จะได้คำตอบที่ถูกต้องเป็น 1 ใน 416 ของคำตอบที่เป็นไปไม่ได้ ถ้ากำหนดจำนวนประชากรเป็น 100 จะเป็นการคำนวณเพียง 1 generation ก็ไม่อาจคาดได้ว่าจะได้คำตอบที่ดี การกำหนดจำนวนประชากรเริ่มต้นและจำนวน generation ทั้งหมดจึงมีผลในการหาคำตอบ

2.3 ตัวอย่างการใช้เจเนติกอัลกอริทึมในการหาคำตอบของฟังก์ชัน

เป็นการประยุกต์ใช้ Gas ในการแก้ปัญหาค่าสูงสุดของฟังก์ชัน $f(x) = x^2$ โดย x เป็นตัวแปรที่มีค่าเปลี่ยนแปลงระหว่าง 0-31

ตารางที่ 2.4 แสดงการคำนวณหาคำตอบของ SGA กับฟังก์ชัน $f(x) = x^2$

หมายเลขสตริง	ประชากรเริ่มต้น (สร้างขึ้นแบบสุ่ม)	ค่า x (unsigned integer)	$f(x) = x^2$	Pselect $f_i/\sum f$	Expected count $f_i/\bar{f}$	Actual Count (จากวงล้อรูเล็ต)
1	01101	13	169	0.14	0.58	1
2	11000	24	576	0.49	1.97	2
3	01000	8	64	0.06	0.22	0
4	10011	19	361	0.31	1.23	1
ผลรวม			1170	1.00	4.00	4.0
ค่าเฉลี่ย			293	0.25	1.00	1.0
ค่าสูงสุด			576	0.49	1.97	2.0

ก) การสุ่มสตริงเริ่มต้นและการรีโปรดัคชัน

เมทติ้งพูลหลังจากการ รีโปรแกรม	สตริงจับคู่ (เลือกแบบสุ่ม)	ตำแหน่งครอสโอเวอร์ (เลือกแบบสุ่ม)	ประชากรใหม่	ค่า x	$f(x) = x^2$
0110 1	2	4	01100	12	144
1100 0	1	4	11001	25	625
11 000	4	2	11011	27	729
10 011	3	2	10000	16	256
ผลรวม					1754
ค่าเฉลี่ย					439
ค่าสูงสุด					729

ข) การครอสโอเวอร์

หมายเหตุ

1. ประชากรเริ่มแรกทั้ง 4 ตัว ในแต่ละตัวได้มาจากการสุ่มโยนเหรียญ 5 ครั้ง (มี 5 บิต)
2. รีโปรแกรมได้มาจากการหมุนวงล้อรูเล็ต
3. ครอสโอเวอร์ได้จากการโยนเหรียญ 2 เหรียญแล้วทำการถอดรหัส ($TT = 00_2 = 0 =$ ตำแหน่งที่ไขว้คือ 1, $HH = 11_2 = 3 =$ ตำแหน่งที่ไขว้คือ 4)
4. ความน่าจะเป็นของการครอสโอเวอร์กำหนดให้เป็นหนึ่ง $p_c = 1.0$
5. ความน่าจะเป็นของมิวเตชันเป็น 0.001 , $p = 0.001$, $expected\ mutation = 5*4*0.001 = 0.2$ ไม่มีค่า $expected\ mutation$ ระหว่างประชากรเดียว

จากผลการทดลองหลังจากผ่านการรีโปรแกรม ครอสโอเวอร์ และมิวเตชัน จะเห็นได้ว่ากระบวนการที่เกี่ยวข้องกับความน่าจะเป็นจะทำให้ค่าสมรรถนะดีขึ้น ค่าความเหมาะสมของประชากรโดยเฉลี่ยมีค่าเพิ่มขึ้นจาก 293 เป็น 439 ในขณะที่ความเหมาะสมสูงสุดมีค่าเพิ่มขึ้นจาก 576 เป็น 729