

บทที่ 3

เจนเนติกอัลกอริทึมและการแก้ปัญหาแบบหลายวัตถุประสงค์

เนื้อหาในบทนี้จะกล่าวถึงทฤษฎีเบื้องต้นของเจนเนติกอัลกอริทึม กับตัวอย่างการคำนวณหาค่าคำตอบของเจนเนติกอัลกอริทึมอย่างง่าย (Simple Genetic Algorithms) เพื่อเป็นพื้นฐานสำหรับการนำเจนเนติกอัลกอริทึมไปใช้ในการแก้ปัญหาการทำ Optimization และตัวอย่างการคำนวณหาค่าคำตอบของเจนเนติกอัลกอริทึมเพื่อนำไปสู่การนำเจนเนติกอัลกอริทึมไปประยุกต์ใช้ในการแก้ปัญหาการจัดผังโรงงานต่อไป

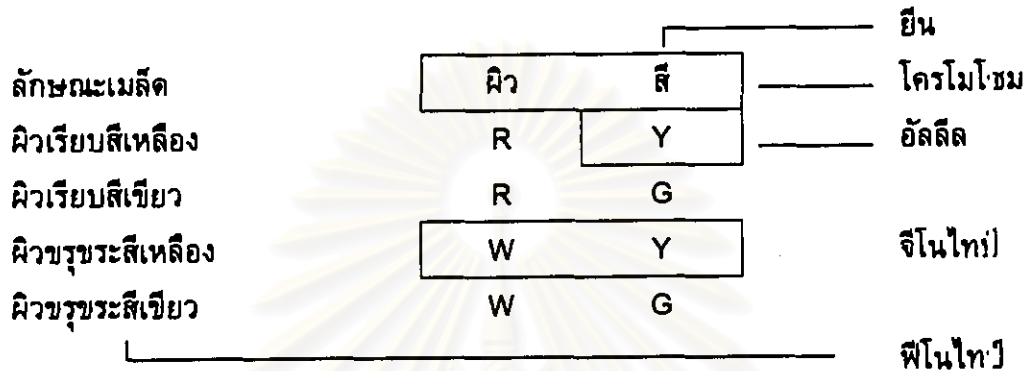
3.1 เจนเนติกอัลกอริทึม

ในปัจจุบันการหาค่าตอบของปัญหาบางประเภท เช่น ปัญหาการจัดสรรทรัพยากรที่มีอยู่อย่างจำกัดเพื่อให้เกิดประสิทธิภาพสูงสุดและปัญหาในการคำนวณต้นทุนต่ำสุด เป็นต้น สามารถหาค่าตอบได้หลายวิธี วิธีการที่ง่ายที่สุดในการหาค่าตอบคือวิธีการทางฮิวริสติกต่างๆ ซึ่งอาจได้คำตอบที่ไม่ดีนัก ในปัจจุบันนักวิทยาศาสตร์ได้นำความรู้เกี่ยวกับทฤษฎีหรือกฎเกณฑ์ทางธรรมชาติมาช่วยในการหาค่าตอบหรือศึกษาวิจัย โดยมีเป้าหมายหลักในการใช้ประโยชน์ของความคงทน (Robustness) ต่อความไม่เที่ยงตรงแม่นยำ (Accuracy) ความไม่แน่นอน (Uncertainty) หรือความคลุมเครือของปัญหา (Vague) หลักการเหล่านี้สามารถพบได้จากวิธีการต่างๆ เช่น ระบบโครงข่ายประสาทเทียม (Neural Network) ฟัซซีลอจิก (Fuzzy Logic) (Zadeh, 1965) และ GAs (Goldberg, 1989) ปัญหาที่พบส่วนใหญ่เป็นปัญหาที่ไม่เที่ยงตรงและคลุมเครือ ซึ่งหาค่าคำตอบที่เที่ยงตรงและมีความแน่นอนสูงมากก็ย่อมมีค่าใช้จ่ายที่สูงมาก ดังนั้นวิธีการที่สามารถแก้ปัญหาที่คลุมเครือโดยได้คำตอบที่ใกล้เคียงสามารถยอมรับได้ ใช้เวลาในการหาค่าคำตอบไม่มากนัก และมีค่าใช้จ่ายพอประมาณ ย่อมดีกว่าวิธีที่ได้ความเที่ยงตรงสูงแต่มีค่าใช้จ่ายที่สูง วิธีการหาค่าตอบที่ตัวอย่างหนึ่งได้แก่วิธีการของ GAs โดยอาศัยทฤษฎีในการถ่ายทอดลักษณะต่างๆทางกรรมพันธุ์ไปสู่ยังลูกหลาน ซึ่งสามารถนำมาพัฒนาใช้ในการหาค่าคำตอบที่ต้องการได้

3.1.1 พันธุศาสตร์กับเจนเนติกอัลกอริทึม

Mendel บิดาแห่งวิชาพันธุศาสตร์ ค้นพบว่าลักษณะต่างๆของสิ่งมีชีวิต เช่น ลักษณะผิวของเมล็ดพืช สีของเมล็ดพืช ฯลฯ ที่ถูกถ่ายทอดไปยังลูกหลานนั้นถูกควบคุมโดยหน่วยควบคุม

ลักษณะที่เรียกว่ายีน (Gene) และลักษณะย่อยของยีนเรียกว่าอัลลีล (Allele) เช่นยีน ควบคุม ลักษณะผิวของเมล็ดจะมีอัลลีลเป็นผิวเรียบและผิวขรุขระ เป็นต้น ซึ่งแต่ละยีนจะเรียงตัวอยู่บน โครโมโซม (Chromosome) ภายในเซลล์ ตำแหน่งของยีนแต่ละยีนบนโครโมโซมเรียกว่า โลกัส (Locus) และแต่ละแบบของชุดยีนเรียกว่า จีโนไทป์ (Genotype) ซึ่งแสดงลักษณะภายนอกที่ ปรากฏ ซึ่งเรียกว่า ฟีนไทป์ (Phenotype) ดังรูปที่ 3.1ก



ก) ลักษณะทางพันธุศาสตร์ของโครโมโซมควบคุมลักษณะของเมล็ดถั่ว ซึ่งมียีนลักษณะของผิว: เมล็ดคือ มี ลักษณะเรียบ (R) หรือ ขรุขระ (W) และยีนลักษณะสีของเมล็ดคือมีสีเหลือง (Y) และสีเขียว (G)

| อักขระ | อักขระ | | ปัญหา |
|----------------|--------|-------|-------|
| สตริง | บิต 1 | บิต 2 | x^2 |
| ค่าอักขระ | 0 | 0 | 0 |
| | 0 | 1 | 1 |
| โครงสร้าง | 1 | 0 | 4 |
| ค่าพารามิเตอร์ | 1 | 1 | 16 |

ค่าคำตอบของปัญหาซึ่งเป็นค่าของฟังก์ชันวัตถุประสงค์

ข) ลักษณะทางเจเนติก แสดงถึงการแก้ปัญหาในการหาค่าสูงสุดของฟังก์ชัน $f(x)=x^2$ โดยที่ x มีค่าอยู่ระหว่าง $[0,4]$ และค่าของ x ถูกแปลงให้อยู่ในรูปไบนารีสตริง รูปที่ 3.1 เปรียบเทียบลักษณะระหว่างเจเนติกอัลกอริทึมกับลักษณะทางพันธุศาสตร์

การแก้ปัญหาทางด้านคณิตศาสตร์ด้วย GAs พารามิเตอร์ต่างๆจะถูกแปลงให้อยู่ในรูปของสตริง (String) หรือโครโมโซมประกอบด้วยอักขระ (Character) หรือ (Bit) แต่ละตำแหน่งของโครโมโซมจะเก็บค่าอักขระหรือค่าของบิตที่แสดงโครงสร้างของแต่ละโครโมโซมที่ให้คำตอบของปัญหาแตกต่างกัน ดังรูปที่ 3.1b) ซึ่งเป็นการประยุกต์ใช้ GAs กับการแก้ปัญหาการหาค่าสูงสุดของ $f(x)=x^2$ โดยที่ x อยู่ในช่วง $[0, 4]$ และสามารถสรุปความหมายทางพันธุศาสตร์เทียบกับ GAs ได้ดังตารางที่ 3.1

ตารางที่ 3.1 เปรียบเทียบคำศัพท์ระหว่างพันธุศาสตร์และเจเนติกอัลกอริทึม

| พันธุศาสตร์ | เจเนติกอัลกอริทึม |
|-----------------------|--|
| โครโมโซม (Chromosome) | สตริง (String) |
| ยีน (Gene) | คุณลักษณะ, บิต (Character, bit) |
| อัลลีล (Allele) | ค่าของคุณลักษณะ, ค่าบิต (Character value, bit value) |
| โลกัส (Locus) | ตำแหน่ง (String position) |
| จีโนไทป์ (Genotype) | โครงสร้าง (Structure) |
| ฟีโนไทป์ (Phenotype) | โครงสร้างคำตอบ (A decode structure) |

Darwin (1859) ได้เสนอความคิดการเกิดสปีชีส์ของสิ่งมีชีวิต (The Origin of Species) โดยเสนอหลักการของวิวัฒนาการที่ผ่านกระบวนการคัดเลือกตามธรรมชาติ แม้ในตอนแรกทฤษฎีจะเป็นที่โต้แย้งกันมากต่อมาก็ได้เป็นที่ยอมรับในหมู่นักวิทยาศาสตร์

- สิ่งมีชีวิตแต่ละชนิดมีแนวโน้มที่จะถ่ายทอดลักษณะของมันไปสู่ลูกหลานของมัน
- ธรรมชาติทำให้สิ่งมีชีวิตมีลักษณะต่าง ๆ กัน
- สิ่งมีชีวิตมีความเหมาะสม ซึ่งมีลักษณะที่เหมาะสมที่สุด มีแนวโน้มที่จะมีลูกหลานมากกว่าสิ่งมีชีวิตที่มีลักษณะไม่เหมาะสม ซึ่งจะทำให้ประชากรอยู่รอดต่อไป
- เมื่อระยะเวลาผ่านไปยาวนาน จะเกิดการกลายพันธุ์ (Variation) ขึ้น และเกิดสปีชีส์ใหม่ที่มีลักษณะเหมาะสมกับระบบนิเวศนั้น

3.1.2 ความหมายของเจเนติกอัลกอริทึม

GAs เป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานมาจากกระบวนการคัดเลือกทางธรรมชาติ (Natural Selection) และ กระบวนการคัดเลือกทางพันธุศาสตร์ (Natural Genetics Selection) โดยการคัดเลือกสตริง (String) ที่มีความเหมาะสมจากกลุ่มของสตริงทั้งหมดด้วยวิธีการสุ่ม จากการนำสตริงเหล่านี้ไปผ่านกระบวนการคัดเลือกสตริงที่มีความเหมาะสม ซึ่งสตริงที่มีความเหมาะสมนี้คือคำตอบที่ดีที่สุดหรือใกล้เคียงคำตอบที่ดีที่สุด GAs ไม่ใช้การสุ่มแบบง่าย ๆ แต่มันเป็นการใช้ข้อมูลในอดีตอย่างมีประสิทธิภาพเพื่อพิจารณาจุดที่จะต้องค้นหาใหม่ โดยคาดหวังว่าสมรรถนะของการค้นหาจะดีขึ้น

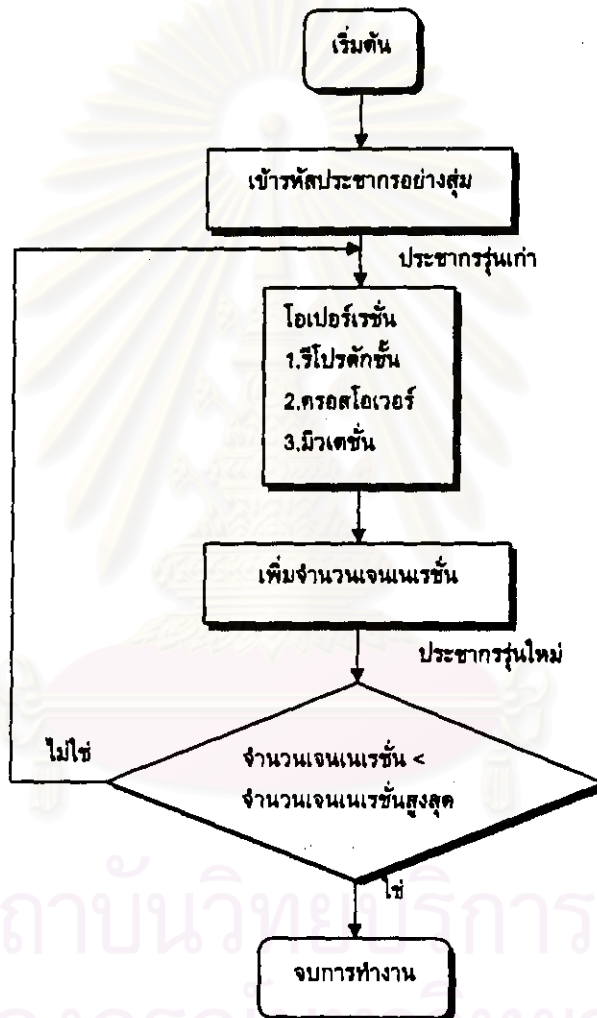
GAs ถูกพัฒนาขึ้นโดย Holland (1975) และคณะ โดยมีเป้าหมายในการวิจัย 2 อย่างคือ ข้อแรก เพื่อสรุปและตัดแปลงการใช้กระบวนการทางธรรมชาติให้ถูกต้องมากที่สุด สองเพื่อออกแบบและสร้างซอฟต์แวร์ที่รักษากลไกที่สำคัญของธรรมชาติ และ GAs แตกต่างกับวิธีการค้นหาและการทำ Optimization แบบอื่นๆ คือ

- GAs ทำงานโดยการเข้ารหัสสตริงเป็นชุดพารามิเตอร์
- GAs เป็นการค้นหาจากทั้งประชากรไม่ใช่ค้นหาจากเพียงตำแหน่งๆเดียว

- GA ใช้ฟังก์ชันเป้าหมายเป็นข้อมูลในการหาค่าตอบ โดยไม่ใช้การหาอนุพันธ์หรือความรู้อื่น
- GAs จะเป็นวิธี Probabilistic ไม่ใช่ Deterministic

3.2 เจเนติกอัลกอริทึมอย่างง่าย (Simple Genetic Algorithms)

ขั้นตอนการทำ GAs อย่างง่าย (Simple Genetic Algorithms) หรือ SGA ดังรูปที่ 3.2



รูปที่ 3.2 ขั้นตอนของ GAs อย่างง่าย

3.2.1 การเข้ารหัสและสร้างประชากรเริ่มต้นอย่างสุ่ม

ขั้นตอนแรกของ GAs คือ การเข้ารหัสหรือแปลงค่าพารามิเตอร์ให้อยู่ในรูปของสตริงที่มีความยาวแน่นอน ซึ่งวิธีการเข้ารหัสนี้ขึ้นอยู่กับรูปแบบของปัญหาแต่ละปัญหา สำหรับ SGA ใช้การเข้ารหัสแบบไบนารี (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 บิตจะได้ ค่าพารามิเตอร์ของ x จะมีค่าอยู่ในช่วง 00000 จนถึง 11111 (0 ถึง 31)

เมื่อกำหนดวิธีการเข้ารหัสแล้ว จำเป็นที่จะต้องสร้างประชากรเริ่มต้น (Initial Population) โดยวิธีการสุ่มเพื่อที่จะผ่านขั้นตอนของ SGA ต่อไป สมมุติว่าสุ่มประชากรเริ่มต้น 4 สดริงได้เป็น

01101

11000

01000

10011

ค่าสดริงของประชากรเริ่มต้นนี้ เกิดจากการสุ่มค่า ทั้งหมด 20 ครั้งหรือ สดริงแต่ละตัว ทำการสุ่ม 5 ครั้ง

3.2.2. ประชากรรุ่นเก่า (Old Population)

ประชากรรุ่นเก่า คือสดริงที่จะถูกคัดเลือกไปเป็นต้นแบบสำหรับสร้างประชากรรุ่นใหม่ (New Population) โดยประชากรรุ่นเก่าชุดแรกคือประชากรเริ่มต้นนั่นเอง

3.2.3. การดำเนินการของ SGA

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

▪ **รีโพรดักชัน** คือกระบวนการที่สดริงแต่ละตัวเลียนแบบค่าฟังก์ชันเป้าหมาย $f(x)$ โดยที่ฟังก์ชันนี้อาจเป็นการวัด ผลตอบแทน ค่ารรถประโยชน์ (Utility function) หรือสิ่งที่ต้องการให้เป็นค่าสูงสุด หรือค่าความเหมาะสม (Fitness) สดริงที่มีความเหมาะสมสูงกว่าก็จะมีโอกาสน่าจะเป็นในการสนับสนุนลูกหลานรุ่นต่อไปสูงด้วย ตัวปฏิบัติการนี้เกิดขึ้นจากกระบวนการคัดเลือกตามธรรมชาติตามทฤษฎีผู้รอดที่มีความเหมาะสม (Survival of Fittest) ของ ชาลส์ ดาร์วิน ประชากรที่มีความเหมาะสมในธรรมชาติจะมีความสามารถในการรอดพ้นจากผู้ล่า โรคภัยไข้เจ็บ อุปสรรคอื่น ๆ ที่ต่อต้านการเจริญเติบโตเป็นผู้ใหญ่และสามารถสืบพันธุ์ต่อไปได้ ส่วนฟังก์ชันเป้าหมายจะเป็นสิ่งที่ใช้พิจารณาว่าสดริงที่สร้างขึ้นจะมีชีวิตอยู่หรือตายจากไป

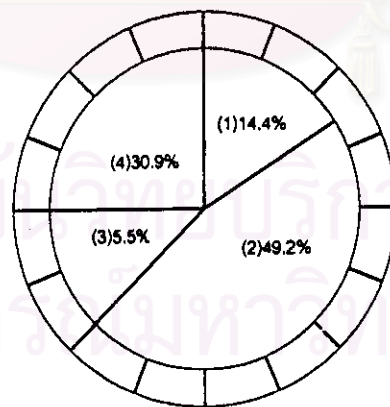
ตัวปฏิบัติการรีโพรดักชันสามารถสร้างขึ้นได้หลายวิธี วิธีที่ง่ายวิธีหนึ่งคือสร้างจากวงล้อรูเล็ตที่มีจำนวนช่องเท่ากับจำนวนประชากรสดริง และขนาดของช่องก็เป็นสัดส่วนกับค่าความ

เหมาะสม ดังรูปที่ 3.3 และค่าความเหมาะสมของฟังก์ชันเป้าหมายของประชากรทั้งสี่แสดงอยู่ในตาราง 3.2

ค่าความเหมาะสมทั้งหมดโดยรวมจะได้ 1170 และค่ารายละเอียดต่างๆแสดงดังในตารางที่ 3.2 แสดงถึงวงล้อรูเล็ตสำหรับการรีโพรดักชัน ซึ่งสร้างจากสัดส่วนของค่าฟิตเนสของสตริงทั้งหมด เช่นสตริงหมายเลข 1 มีค่าความเหมาะสมเป็น 169 หรือ 14.4% (169/1170) ของค่าฟิตเนสโดยรวมของทั้งประชากร ในการทำการรีโพรดักชันจะหมุนวงล้อเป็นจำนวน 4 ครั้ง หรือเท่ากับจำนวนสตริง เช่นสตริงหมายเลข 1 มีค่าเป็น 169 คิดเป็น 14.4% ของค่าความเหมาะสมทั้งหมด ดังนั้นเมื่อหมุนวงล้อ 1 ครั้งก็จะมีโอกาสที่จะถูกเลือกเท่ากับ 0.144 ในการหมุนวงล้อแต่ละครั้งจะได้ตัวแทนในการสืบพันธุ์ (Reproduction Candidate) สตริงที่มีความเหมาะสมสูงจะถูกคัดเลือกสำหรับการสืบพันธุ์การรีโพรดักชันสำหรับสตริงลูกหลานในรุ่นต่อไป เมื่อสตริงมีรูปร่างที่แน่นอนแล้วก็จะถูกส่งไปเข้าเมทติงพูล (Mating Pool) เเงื่อที่จะผ่านกระบวนการของตัวปฏิบัติการอื่นต่อไป

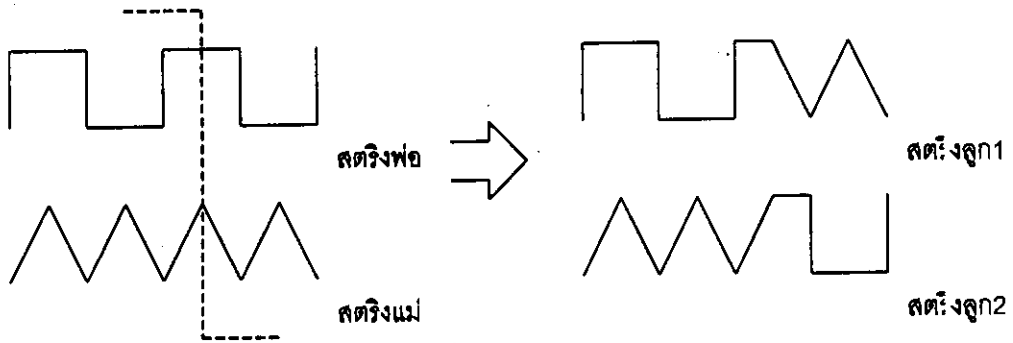
ตารางที่ 3.2 กลุ่มประชากรตัวอย่างและค่าความเหมาะสม

| No. | สตริง | ค่าความเหมาะสม | % โดยรวม |
|-----|-------|----------------|----------|
| 1 | 01101 | 169 | 14.40 |
| 2 | 11000 | 576 | 49.20 |
| 3 | 01000 | 64 | 5.50 |
| 4 | 10011 | 361 | 30.90 |
| รวม | | 1170 | 100.00 |



รูปที่ 3.3 การรีโพรดักชันอย่างง่ายด้วยวิธีการใช้วงล้อรูเล็ตที่มีขนาดของแต่ละช่องเป็นสัดส่วนกับค่าความเหมาะสม

▪ **ครอสโอเวอร์(Crossover)** หลังจากประชากรทั้งหมดผ่านกระบวนการรีโพรดักชันแล้ว จะทำการจับคู่สมาชิกในเมทติงพูลหรือกลุ่มประชากรทั้งหมดอย่างสุ่มและทำการไขว้สลับค่าที่อยู่หลังตำแหน่งที่เลือกไว้จากการสุ่มหรือ ทำการแลกเปลี่ยนสวนกัน



รูปที่ 3.4 การครอสโอเวอร์อย่างง่ายเพื่อให้เห็นถึงการเปลี่ยนแปลงสตริง และการแลกเปลี่ยนข่าวสารโดยเลือกตำแหน่งไขว้แบบสุ่ม

การเลือกตำแหน่งที่จะทำการครอสโอเวอร์ จะทำโดยการสุ่มค่าที่เป็นจำนวนเต็ม ตำแหน่งที่ k ช่วงของสตริงที่เลือกจะอยู่ในช่วง $[2, t-1]$ โดยที่ t คือตำแหน่งสุดท้ายของสตริงใหม่ทั้งสองก็จะมี การสลับอักขระตั้งแต่ตำแหน่งที่ $k+1$ จนถึง t ยกตัวอย่างเช่น พิจารณาสตริง A_1, A_2 จากประชากรเริ่มต้น

$$\begin{array}{l} A_1 = 0110 \\ A_2 = 1100 \end{array} \begin{array}{l} | \\ | \\ | \\ | \\ \times \\ \times \\ \times \\ \times \\ | \\ | \end{array} \begin{array}{l} 1 \\ 0 \\ 0 \\ 1 \end{array}$$

สมมุติว่าเลือกจำนวนสุ่มระหว่าง 1 ถึง 4 และได้ค่า $k = 4$ (แสดงโดยใช้สัญลักษณ์ “|” แทนการแยก) ผลของการครอสโอเวอร์สตริงที่เป็นประชากรรุ่นใหม่จะมีสัญลักษณ์ “'”

$$\begin{array}{l} A'_1 = 01100 \\ A'_2 = 11001 \end{array}$$

▪ **มิวเตชัน(Mutation)** มิวเตชันเป็นสิ่งที่จำเป็นถึงแม้ว่าวิโปรดักชันและครอสโอเวอร์ช่วยให้การค้นหาเป็นไปอย่างมีประสิทธิภาพในบางครั้งก็มีการสูญเสียส่วนที่สำคัญไป (ค่า 1 หรือ 0 ในบางตำแหน่ง) การมิวเตชันจะป้องกันส่วนที่สูญเสียที่ไม่อาจเรียกคืนได้ (Irrecoverable Loss) ในบางครั้งการหาค่าตอบของ SGA ค่าตอบอาจติดอยู่ใน Local Optima การมิวเตชันด้วยอัตราส่วนที่เหมาะสมจะทำให้ค่าตอบสามารถหลุดออกจาก Local Optima หรืออาจกล่าวได้ว่าโอเพอร์เรเตอร์ของการมิวเตชันเป็นการเปลี่ยนแปลงค่าตำแหน่งสตริงแบบสุ่ม จากปัญหาที่พิจารณาค่าจะเปลี่ยนแปลงจาก 0 เป็น 1 หรือ 1 เป็น 0 โดยการเลือกตำแหน่งที่จะทำการมิวเตชันอย่างสุ่ม อัตราการมิวเตชันในธรรมชาติจะมีค่าค่อนข้างต่ำ ในการนำไปใช้งานจะต้องมีการพิจารณาอย่างเหมาะสม

3.2.4 ประชากรรุ่นใหม่ (New population)

สตริงทั้งหมดที่ได้จากกระบวนการของ GAs เรียกว่าประชากรรุ่นใหม่หรือเจนเนอเรชัน (Generation) รุ่นใหม่ซึ่งจะกลายเป็นประชากรรุ่นเก่า สำหรับการดำเนินการต่อไป กระบวนการของ SGA จะทำซ้ำไปเรื่อยๆ จนกว่าจำนวนเจนเนอเรชันจะมากกว่าจำนวนเจนเนอเรชันที่กำหนดไว้สูงสุด

Surech (1995) ได้พิจารณาถึงการหาขนาดของประชากร จากอัตราส่วนของวิธีการที่ทั้งหมดของคำตอบที่เป็นไปไม่ได้และอัตราส่วนของวิธีการที่ทั้งหมดของคำตอบที่เป็นไปได้

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{n^n}{n!} &\approx \lim_{n \rightarrow \infty} \frac{(2\pi)^{1/2} \left(\frac{n}{e}\right)^n}{n^n} \\ &= \lim_{n \rightarrow \infty} \frac{(2\pi)^{1/2}}{e^n} = 0 \end{aligned} \quad (3.1)$$

จากสมการที่ (3.1) ให้

n คือขนาดของปัญหา

m คือจำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงที่เป็นไปได้

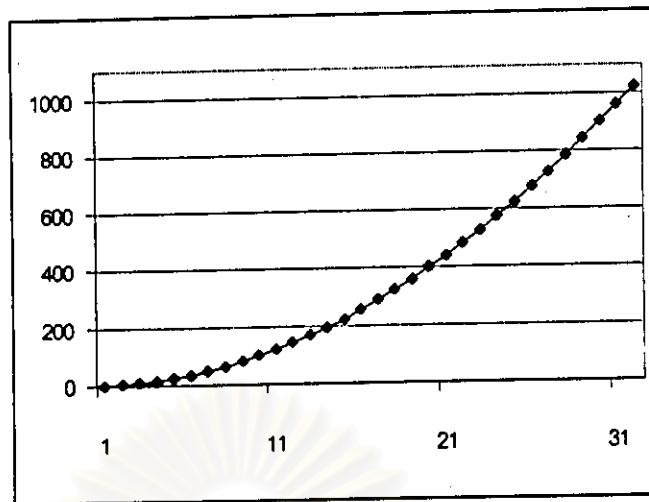
n^n คือจำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงที่เป็นไปไม่ได้ (n, n, \dots, n)

สามารถสรุปได้ว่า ความน่าจะเป็นของการสร้างประชากรคำตอบอย่างสุ่ม มีค่าเป็นศูนย์เมื่อ n มีค่ามากขึ้น สมมติว่า n มีค่าเป็น 8 ดังนั้น $8! / 8^8 = 1 / 416.1 = 2.403 \times 10^{-3}$ หรืออาจกล่าวได้ว่า โอกาสที่จะได้คำตอบที่ถูกต้องเป็น 1 ใน 416 ของคำตอบที่เป็นไปไม่ได้ ถ้ากำหนดจำนวนประชากรเป็น 100 และทำการคำนวณเพียงแค่ 1 เจนเนอเรชันก็ไม่อาจคาดได้ว่าจะได้คำตอบที่ดี การกำหนดจำนวนประชากรเริ่มต้นและจำนวนเจนเนอเรชันทั้งหมดจึงมีผลในการหาคำตอบ

3.3 ตัวอย่างการใช้เงินเนติกอัลกอริทึมในการหาคำตอบของฟังก์ชัน

เนื้อหาในส่วนนี้จะเป็นการประยุกต์ใช้ GAs ในการแก้ปัญหา Optimization หาค่าสูงสุดของฟังก์ชัน $f(x) = x^2$ ที่ละขั้นตอน โดย x เป็นตัวแปรที่มีค่าเปลี่ยนแปลงระหว่าง 1 ถึง 31 ดังรูปที่ 3.5 แสดงถึงลักษณะฟังก์ชัน $f(x)$ สำหรับปัญหานี้ตัวแปร x จะถูกเข้ารหัสให้เป็นไบนารีที่มีความยาวสตริง 5 บิต

วิธีการทำเริ่มจากเลือกประชากรแรกขึ้นมาอย่างสุ่ม โดยประชากรเริ่มแรกจะได้มาจากการโยนเหรียญ 20 ครั้ง จากตาราง 3.3 จะเห็นได้ว่าสตริงหมายเลข 3 ซึ่งมีค่าเป็น 01000 (นำมาเข้าแปลงเป็นเลขฐานสิบ คือ $2^3 = 8$) จากนั้นก็จะแปลงให้อยู่ในฟังก์ชันเป้าหมาย $f(x) = x^2$ จะได้ค่าเป็น 64 สำหรับค่า x และ $f(x)$ อื่นๆก็คิดในลักษณะเดียวกัน



รูปที่ 3.5 ฟังก์ชันวัตถุประสงค์ $f(x) = x^2$

ตารางที่ 3.3 การคำนวณหาค่าต่อของ SGA กับฟังก์ชัน $f(x) = x^2$

3.3 ก) การสุ่มสตริงเริ่มต้นและการรีโปรดักชัน

| หมายเลขสตริง | ประชากรเริ่มต้น (สร้างขึ้นแบบสุ่ม) | ค่า x (unsigned integer) | $f(x) = x^2$ | Pselect $f/\sum f$ | Expected count f/\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 | 6 | 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 |

3.3 ข) การครอสโอเวอร์

| เมทติ้งพูลหลังจากการ รีโปรดักชัน | 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 |

หมายเหตุ

- ประชากรเริ่มแรกทั้งสี่ตัว ในแต่ละตัวได้มาจากการสุ่มโยนเหรียญ 5 ครั้ง (มี 5 บิต)

- รีโปรดักชันได้จากการหมุนวงล้อรูเล็ต
- ครอสโอเวอร์ได้จากการโยนเหรียญสองเหรียญแล้วทำการถอดรหัส ($TT = 0$)₂ = 0 = ตำแหน่งที่ไขว้คือ 1, HH = 11₂ = 3 = ตำแหน่งที่ไขว้คือ 4)
- ความน่าจะเป็นของครอสโอเวอร์กำหนดให้เป็นหนึ่ง $p_c = 1.0$
- ความน่าจะเป็นของมิวเตชันเป็น 0.001, $p = 0.001$, expected mutation = $5 \cdot 4 \cdot 0.001 = 0.2$ ไม่มีค่า expected mutation ระหว่างประชากรเดียว

ประชากรรุ่นต่อไป จะเริ่มต้นกระบวนการรีโปรดักชันจากเมทดิงพูล โดยการหมุนวงล้อรูเล็ต 4 ครั้ง ได้สตริงหมายเลข 1 และ 4 ได้รับการคัดเลือกไปยังรุ่นต่อไป 1 ครั้ง สตริง 2 ได้รับการคัดเลือกไปยังรุ่นต่อไป 2 ครั้ง สตริง 3 ไม่ได้ได้รับการคัดเลือกไปยังรุ่นต่อไปเลย เมื่อเปรียบเทียบจำนวนครั้งที่ถูกคัดเลือกที่คาดหวัง หรือ Expected Count (หาได้จาก f_i / \bar{f}) กับจำนวนครั้งที่ถูกคัดเลือกจริงจะเห็นได้ว่ามีค่าใกล้เคียงกัน ค่าที่ดีที่สุดจะมีโอกาสที่จะถูกคัดเลือกมากกว่า ส่วนค่าที่ไม่ดีก็จะตายจากไป

ขั้นตอนต่อไปคือการครอสโอเวอร์ ซึ่งจะต้องมีการจับคู่กันระหว่างสตริง โดยมีสองขั้นตอนคือ (1) สตริงจะถูกจับคู่อย่างสุ่มโดยใช้วิธีการโยนเหรียญจับคู่ (2) สตริงจะทำการครอสโอเวอร์โดยการโยนเหรียญเพื่อเลือกตำแหน่งที่จะไขว้ (Crossing sites) เมื่อพิจารณาตาราง 3.3 ข อีกครั้ง จะเห็นได้ว่าการสุ่มจับคู่ในเมทดิงพูล สตริงหมายเลข 2 จะจับคู่กับสตริงหมายเลข 1 และมีตำแหน่งการไขว้คือ 4 สตริงทั้งสองคือ 01101 และ 11000 เมื่อทำการไขว้จะได้สตริงตัวใหม่คือ 01100 และ 11001 สตริงที่เหลือในเมทดิงพูลจะทำการไขว้กันในตำแหน่งที่สองดังแสดงในตารางที่ 3.3 ข

ตัวปฏิบัติการสุดท้ายคือมิวเตชันซึ่งจะเปลี่ยนค่าเป็นบิตต่อบิต สมมุติความน่าจะเป็นของการมิวเตชันในการทดสอบเป็น 0.001 ตำแหน่งที่จะเปลี่ยนแปลงทั้งหมดมี 20 บิต (ได้จากจำนวนสตริง*จำนวนบิตของสตริงแต่ละตัว $5 \cdot 4 = 20$) เพราะฉะนั้นตำแหน่งบิตที่จะมิวเตชันของประชากรรุ่นนี้คือ $20 \cdot 0.001 = 0.02$ บิต จากการคำนวณจะเห็นได้ว่าไม่มีบิตใดต้องทำการมิวเตชันสำหรับค่าความน่าจะเป็นนี้ นั่นก็คือไม่มีบิตใดที่จะต้องเปลี่ยนค่าจาก 1 เป็น 0 หรือ 0 เป็น 1 สำหรับประชากรรุ่นนี้ แต่สมมติว่าถ้าตำแหน่งบิตที่จะมิวเตชันของประชากรรุ่นนี้คือ 5 ดังนั้นตำแหน่งบิตที่ 5 จะต้องทำการเปลี่ยนค่าจาก 0 เป็น 1 หรือ 1 เป็น 0

หลังจากผ่านการรีโปรดักชัน ครอสโอเวอร์และมิวเตชัน ประชากรรุ่นใหม่ก็พร้อมที่จะถูกทดสอบ โดยทำการเข้ารหัสสตริงใหม่คำนวณหาค่า x และค่าฟังก์ชัน $f(x)$ ตารางที่ 3.3 ข. แสดงถึงผลจากการทดลองจะเห็นได้ว่ากระบวนการที่เกี่ยวข้องกับความน่าจะเป็นจะทำให้ค่าสมรรถนะดีขึ้น ค่าความเหมาะสมของประชากรโดยเฉลี่ยมีค่าเพิ่มขึ้นจาก 293 เป็น 439 ในขณะที่ค่าความเหมาะสมสูงสุดมีค่าเพิ่มขึ้นจาก 576 เป็น 729 ถึงแม้ว่ากระบวนการสุ่มจะช่วยทำให้ค่าต่างๆสูงขึ้นแต่ค่าต่างๆที่เพิ่มขึ้นเหล่านี้ไม่ใช่ความบังเอิญ ค่าสตริงที่ดีที่สุดของประชากรเริ่มแรกคือ (11000) จะมีการเลียนแบบ 2 ครั้งเนื่องจากเป็นค่าที่สูงเกินกว่าค่าเฉลี่ย เมื่อรวมกับค่า

สตริงตัวต่อไป (10011) แบบสุ่มและทำการไขว้แบบสุ่มในตำแหน่งที่สองก็จะได้ผลลัพธ์เป็น (11011) ซึ่งก็จะเป็นค่าที่ดีเช่นกัน

ค่าพารามิเตอร์ของ SGA มีความสำคัญอย่างมาก ค่าพารามิเตอร์เหล่านี้ในบางครั้งจำเป็นต้องมีการปรับเปลี่ยนไปตามรูปแบบของปัญหาเพื่อให้ได้คำตอบที่ดี แต่ในบางครั้งก็ไม่อาจที่จะหาคำตอบที่ดีได้เนื่องจาก (Michalewicz, 1992)

- การเข้ารหัสของปัญหาผิดพลาด ทำให้ GAs หาคำตอบผิดพลาด
- ขีดจำกัดของจำนวนประชากร ในทางทฤษฎีแล้วมีค่าเป็นอนันต์
- ขีดจำกัดของจำนวนเจนเนอเรชัน ในทางทฤษฎีแล้วมีค่าเป็นอนันต์

ค่าพารามิเตอร์เหล่านี้ไม่สามารถกำหนดให้เป็นอนันต์ได้ในทางปฏิบัติเนื่องจากข้อจำกัดต่างๆของคอมพิวเตอร์

3.4 ปัญหาแบบหลายวัตถุประสงค์

สามารถให้นิยามของการแก้ปัญหาแบบหลายวัตถุประสงค์ (Multiobjective Optimization) ได้ดังนี้ (Osyczka, 1985)

“คำตอบของปัญหาลักษณะนี้ประกอบขึ้นจากปัจจัยต่างๆที่อยู่ภายใต้ข้อจำกัด ซึ่งสามารถให้ผลลัพธ์ที่ดีในทุกวัตถุประสงค์ ซึ่งปัจจัยต่างๆนั้นอาจมีความขัดแย้งกันอยู่ในตัวเอง โดยคำว่า “สามารถให้ผลลัพธ์ที่ดีผลลัพธ์ที่ดีในทุกวัตถุประสงค์” นั้นอาจจะหมายถึง ค่าตอบที่ได้ นั้นสามารถตอบสนองวัตถุประสงค์ต่างๆในระดับที่เจ้าของปัญหายอมรับได้”

3.5 วิธีการแก้ปัญหาแบบหลายวัตถุประสงค์

วิธีการแก้ปัญหาแบบหลายวัตถุประสงค์นั้นมีหลายวิธี สำหรับการวิจัยนี้ ได้ใช้วิธีการให้น้ำหนัก (Weight Sum Approach) ซึ่งเป็นวิธีที่ใช้งานง่ายและมีประสิทธิภาพดี

- วิธีการให้น้ำหนัก (Weight Sum Approach)

วิธีการนี้จะทำการรวมวัตถุประสงค์ต่างๆเข้าเป็นวัตถุประสงค์เดียว โดยทำการเพิ่มสัมประสิทธิ์น้ำหนักของแต่ละวัตถุประสงค์ สามารถแสดงดังสมการ(3.2)

$$\min/\max \sum_{i=1}^k w_i f_i(\bar{x}) \quad (3.2)$$

$f_i(x)$ คือ ค่าฟังก์ชันวัตถุประสงค์ต่างๆ

w_i คือค่าสัมประสิทธิ์ที่ให้แต่ละวัตถุประสงค์ ซึ่ง

$$\sum_{i=1}^k w_i = 1 \quad (3.3)$$

วิธีการนี้มีข้อดีคือใช้งานง่าย ไม่ซับซ้อน แต่ก็มีข้อเสียคือ กำหนดน้ำหนักในแต่ละวัตถุประสงค์ให้เหมาะสมได้ยาก ดังนั้นเพื่อให้การกำหนดน้ำหนักมีความถูกต้อง ผู้กำหนดนั้นควรมีความรู้ ความเข้าใจในปัญหา และทราบถึงความจำเป็นในแต่ละวัตถุประสงค์เป็นอย่างดี

3.6 สรุป

GAs เป็นวิธีการค้นหาค่าตอบวิธีหนึ่ง โดยมีพื้นฐานจากกระบวนการคัดเลือกทางธรรมชาติ ข้อดีของ GAs เมื่อเปรียบเทียบกับวิธีการค้นหาแบบอื่นคือมีความคงทนต่อความไม่เที่ยงตรง แม่นยำและความไม่แน่นอนหรือคลุมเครือของปัญหา และสามารถควบคุมได้ โดยมีความน่าเชื่อถือสูงและค่าใช้จ่ายต่ำ

วิธีการค้นหาของ GAs จะแตกต่างกับวิธีการค้นหาและการทำ Optimization แบบอื่นๆ คือ

- GAs จะใช้งานโดยการเข้ารหัสสตริงเป็นชุดพารามิเตอร์
- GAs เป็นการค้นหาจากทั้งประชากรไม่ใช่เพียงตำแหน่ง ๆ เดียว
- GAs ใช้ฟังก์ชันเป้าหมายเป็นข้อมูลในการหาค่าตอบ โดยไม่ใช้การหาอนุพันธ์หรือความรู้อื่นๆ
- GAs จะเป็นวิธี Probabilistic ไม่ใช่ Deterministic

โอเปอร์เรเตอร์ต่างๆของ GAs ได้แก่

- รีโพรดักชัน คือกระบวนการคัดเลือกสตริงที่มีความเหมาะสมสูงเพื่อเป็นค่าตอบเริ่มต้นให้กับประชากรรุ่นต่อไป โดยอาศัยทฤษฎีของ ชาร์ล ดาร์วินที่ว่า สิ่งมีชีวิตที่แข็งแรงกว่ามีโอกาสอยู่รอดในสภาวะนั้นๆ ได้มากกว่า
- การครอสโอเวอร์ คือ กระบวนการสร้างสตริงลูกหลานใหม่ จากสตริงพ่อแม่
- การมิวเตชันคือ กระบวนการที่ช่วยปรับปรุงสตริงให้ดีขึ้นหรืออาจจะเลวลง โดยการเปลี่ยนแปลงค่าในบางตำแหน่งของสตริง เพื่อให้เกิดสตริงใหม่

พารามิเตอร์ต่างๆของ GAs ได้แก่

- จำนวนประชากร (Population size)
- จำนวนเจนเนอเรชัน (Number of generation)
- วิธีการรีโพรดักชัน (Reproduction type)

- วิธีการครอสโอเวอร์(Crossover type)และค่าความน่าจะเป็นของการครอสโอเวอร์ (Probability of Crossover, P_c)
- วิธีการการมิวเตชัน(Mutation type)และค่าความน่าจะเป็นของการมิวเตชัน (Probability of Mutation, P_m)

สามารถประยุกต์เจเนติกอัลกอริทึมเพื่อแก้ปัญหาแบบหลายวัตถุประสงค์ได้ โดยวิธีการที่นิยมใช้และได้ผลดีวิธีหนึ่งคือการให้น้ำหนักในแต่ละปัจจัย (Weight Sum Approach)



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย