บทที่ 3

ทฤษฎีเกี่ยวกับเจนเนติกอัลกอริทึม

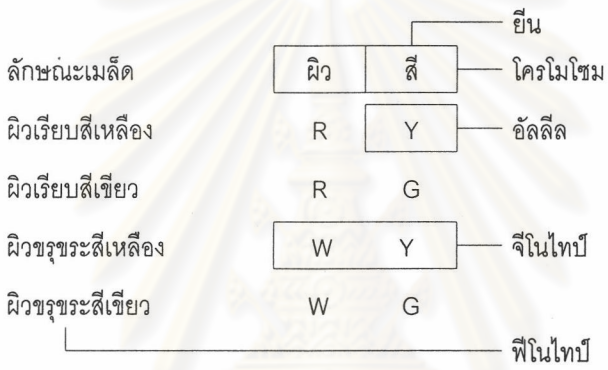
ในการนำเจนเนติกอัลกอริทึม (Genetic Algorithms: GAs) ไปประยุกต์ในการจัดตารางการผลิตที่มีเวลาในการปรับตั้งขึ้นอยู่กับผลิตภัณฑ์ก่อนหน้านั้น จำเป็นต้องทราบถึงทฤษฎีเบื้องต้นของเจนเนติกอัลกอริทึมก่อน ดังนั้นเนื้อหาในบทนี้จะกล่าวถึงทฤษฎีเบื้องต้นของเจนเนติกอัลกอริทึมและตัวอย่างการคำนวณหาค่าคำตอบของเจนเนติกอัลกอริทึมอย่างง่าย เพื่อเป็นพื้นฐานสำหรับการนำเจนเนติกอัลกอริทึมไปใช้ในการแก้ปัญหาการค้นหาคำตอบที่ดีที่สุด

3.1 เจนเนติกอัลกอริทึม

ในปัจจุบันการหาคำตอบของปัญหาบางประเภท เช่น ปัญหาการจัดสรรทรัพยากรที่มีอยู่อย่างจำกัดเพื่อให้เกิดประสิทธิภาพสูงสุดและปัญหาในการคำนวณต้นทุนต่ำสุด เป็นต้น สามารถหาคำตอบได้หลายวิธี วิธีการที่ง่ายที่สุดในการหาคำตอบคือวิธีการทางฮิวริสติกต่างๆ ซึ่งอาจได้คำตอบที่ไม่ดีนัก นักวิทยาศาสตร์จึงได้นำความรู้เกี่ยวกับทฤษฎีหรือกฎเกณฑ์ทางธรรมชาติมาช่วยในการหาคำตอบหรือศึกษาวิจัย โดยมีเป้าหมายหลักในการใช้ประโยชน์ของความคงทน (Robustness) ต่อความไม่เที่ยงตรงแม่นยำ (Accuracy) ความไม่แน่นอน (Uncertainty) หรือความคลุมเครือของปัญหา (Vague) หลักการเหล่านี้สามารถพบได้จากวิธีการต่างๆ เช่น ระบบโครงข่ายประสาทเทียม (Neural Network) ฟัซซี่ลอจิก (Fuzzy Logic) และ เจนเนติกอัลกอริทึม (Goldberg, 1989) ปัญหาที่พบส่วนใหญ่เป็นปัญหาที่ไม่เที่ยงตรงและคลุมเครือ ซึ่งหากต้องการคำตอบที่เที่ยงตรงและมีความแน่นอนสูงมากก็ย่อมมีค่าใช้จ่ายที่สูงมาก ดังนั้นวิธีการที่สามารถแก้ปัญหาที่คลุมเครือโดยได้คำตอบที่ใกล้เคียงสามารถยอมรับได้ ใช้เวลาในการหาคำตอบไม่มากนัก และมีค่าใช้จ่ายพอประมาณ ย่อมดีกว่าวิธีที่ได้ความเที่ยงตรงสูงแต่มีค่าใช้จ่ายที่สูง วิธีการหาคำตอบที่ดีอย่างหนึ่งได้แก่วิธีการของเจนเนติกอัลกอริทึมโดยอาศัยทฤษฎีในการถ่ายทอดลักษณะต่างๆทางกรรมพันธ์ไปสู่ยังลูกหลาน ซึ่งสามารถนำมาพัฒนาใช้ในการหาคำตอบที่ต้องการได้

3.1.1 พันธุศาสตร์กับเจนเนติกอัลกอริทึม

Mendel บิดาแห่งวิชาพันธุศาสตร์ ค้นพบว่าลักษณะต่างๆของสิ่งมีชีวิต เช่น ลักษณะผิวของเมล็ดพืช สีของเมล็ดพืช ฯลฯ ที่ถูกถ่ายทอดไปยังลูกหลานนั้นถูกควบคุมโดยหน่วยควบคุมลักษณะที่เรียกว่ายีน (Gene) และลักษณะย่อยของยีนเรียกว่าอัลลีล (Allele) เช่นยีน ควบคุมลักษณะผิวของเมล็ดจะมีอัลลีลเป็นผิวเรียบและผิวขรุขระ เป็นต้น ซึ่งแต่ละยีนจะเรียงตัวอยู่บนโครโมโซม (Chromosome) ภายในเซลล์ ตำแหน่งของยีนแต่ละยีนบนโครโมโซมเรียกว่า โลกัส (Logus) และแต่ละแบบของชุดยีนเรียกว่า จีโนไทป์ (Genotype) ซึ่งแสดงลักษณะภายนอกที่ปรากฏ ซึ่งเรียกว่า ฟีโนไทป์ (Phenotype) ดังรูปที่ 3.1ก



ก) ลักษณะทางพันธุศาสตร์ของโครโมโซมควบคุมลักษณะของเมล็ดถั่ว ซึ่งมียีนลักษณะของผิวเมล็ดคือ มีลักษณะเรียบ (R) หรือ ขรุขระ (W) และยีนลักษณะสีของเมล็ดคือมีสีเหลือง (Y) และสีเขียว (G)

ข) ลักษณะทางเจนเนติก แสดงถึงการแก้ปัญหาในการหาค่าสูงสุดของฟังก์ชัน $f(x)=x^2$ โดยที่ x มีค่าอยู่ระหว่าง [0,4] และค่าของ x ถูกแปลงให้อยู่ในรูปไบนารี่สตริง

***รูปที่ 3.1** เปรียบเทียบลักษณะระหว่างเจนเนติกอัลกอริทึมกับลักษณะทางพันธุศาสตร์*

ตารางที่ 3.1 เปรียบเทียบคำศัพท์ระหว่างพันธุศาสตร์และเจนเนติกอัลกอริทึม

พันธุศาสตร์	เจนเนติกอัลกอริทึม
โครโมโซม (Chromosome)	สตริง (String)
ยีน (Gene)	คุณลักษณะ, บิท (Character, bit)
อัลลีล (Allele)	ค่าของคุณลักษณะ, ค่าบิท(Character value, bit value)
โลกัส (Locus)	ตำแหน่ง (String position)
จีโนไทป์ (Genotype)	โครงสร้าง (Structure)
พีโนไทป์ (Phenotype)	โครงสร้างคำตอบ (A decode structure)

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

Darwin (1859) ได้เสนอความคิดการเกิดสปีชีส์ของสิ่งมีชีวิต (The Origin of Species) โดยเสนอหลักการของวิวัฒนาการที่ผ่านกระบวนการคัดเลือกตามธรรมชาติ แม้ในตอนแรกทฤษฎีจะเป็นที่โต้แย้งกันมากต่อมาก็ได้เป็นที่ยอมรับในหมู่นักวิทยาศาสตร์ (Winston, 1992)

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

3.1.2 ความหมายของเจนเนติกอัลกอริทึม

GAs เป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานมาจากกระบวนการคัดเลือกทางธรรมชาติ (Natural Selection) และ กระบวนการคัดเลือกทางพันธุศาสตร์ (Natural Genetics

Selection) โดยการคัดเลือกสตริง (String) ที่มีความเหมาะสมจากกลุ่มของสตริงทั้งหมดด้วยวิธีการสุ่ม จากการนำสตริงเหล่านี้ไปผ่านกระบวนการคัดเลือกสตริงที่มีความเหมาะสม ซึ่งสตริงที่มีความเหมาะสมนี้คือคำตอบที่ดีที่สุดหรือใกล้เคียงคำตอบที่ดีที่สุดเจนเนติกอัลกอริทึมไม่ใช่การสุ่มแบบง่ายๆ แต่มันเป็นการใช้ข้อมูลในอดีตอย่างมีประสิทธิภาพเพื่อพิจารณาจุดที่จะต้องค้นหาใหม่โดยคาดหวังว่าสมรรถนะของการค้นหาจะดีขึ้น

เจนเนติกอัลกอริทึมถูกพัฒนาขึ้นโดย Holland (1975) และคณะ โดยมีเป้าหมายในการวิจัย 2 อย่าง คือ ข้อแรก เพื่อสรุปและดัดแปลงการใช้กระบวนการทางธรรมชาติให้ถูกต้องมากที่สุด สองเพื่อออกแบบและสร้างซอฟแวร์ที่รักษากลไกที่สำคัญของธรรมชาติ และเจนเนติกอัลกอริทึมแตกต่างกับวิธีการค้นหาและการทำ Optimization แบบอื่นๆ คือ

- เจนเนติกอัลกอริทึมทำงานโดยการเข้ารหัสสตริงเป็นชุดพารามิเตอร์
- เจนเนติกอัลกอริทึมเป็นการค้นหาจากทั้งประชากรไม่ใช่ค้นหาจากเพียงตำแหน่งเดียว
- เจนเนติกอัลกอริทึมใช้ข่าวสารที่เป็นผลลัพธ์ (ฟังก์ชันเป้าหมาย) โดยไม่ใช้การอนุพันธ์หรือความรู้อื่นๆ
- เจนเนติกอัลกอริทึมจะเป็นวิธี Probabilistic ไม่ใช่ Deterministic

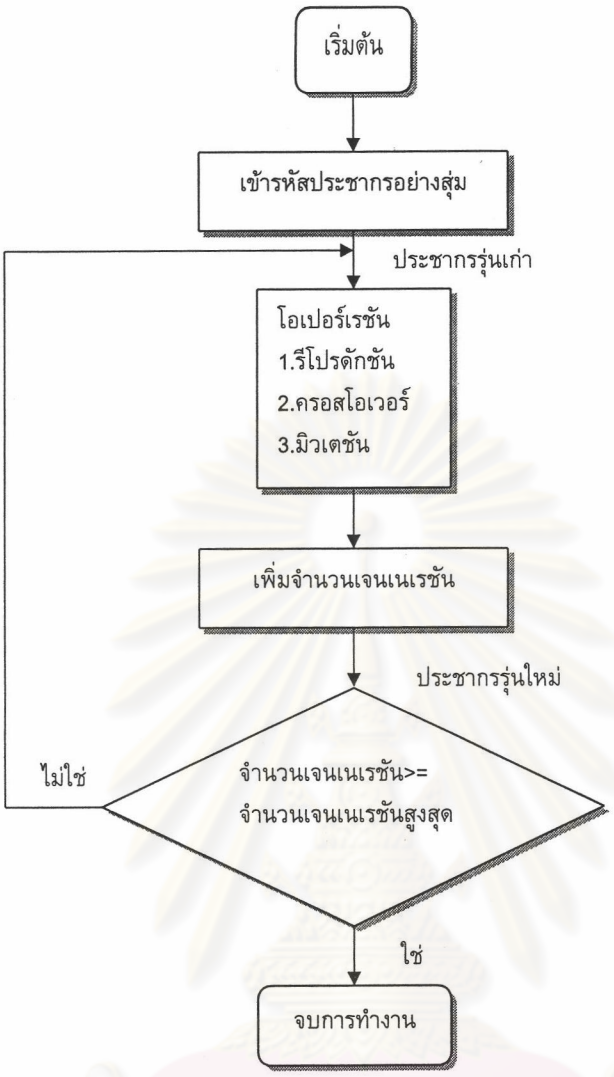
3.2 เจนเนติกอัลกอริทึมอย่างง่าย (Simple Genetic Algorithms)

ขั้นตอนการทำเจนเนติกอัลกอริทึมอย่างง่าย (Simple Genetic Algorithms: SGA) ดังรูปที่ 3.2 มีรายละเอียดดังต่อไปนี้

3.2.1 การเข้ารหัสและสร้างประชากรเริ่มต้นอย่างสุ่ม

ขั้นตอนแรกของเจนเนติกอัลกอริทึมคือ การเข้ารหัสหรือแปลงค่าพารามิเตอร์ให้อยู่ในรูปของสตริงที่มีความยาวแน่นอน ซึ่งวิธีการเข้ารหัสนี้ขึ้นอยู่กับรูปแบบของปัญหาแต่ละปัญหา สำหรับ SGA ใช้การเข้ารหัสแบบไบนารี (Binary Coding)

ตัวอย่างเช่น ต้องการหาค่าสูงสุดของฟังก์ชัน $f(x) = x^2$ โดยที่ x มีค่าอยู่ระหว่าง [0,31] ในที่นี่ฟังก์ชันวัตถุประสงค์ (Objective Function) คือ f(x) หรือ x^2 ซึ่งวิธีการเข้ารหัสแบบไบนารี โดยแปลงค่าพารามิเตอร์ x ให้อยู่ในรูปไบนารี 5 บิทจะได้ ค่าพารามิเตอร์ของ x จะมีค่าอยู่ในช่วง 00000 จนถึง 11111 (0 ถึง 31)



รูปที่ 3.2 *ขั้นตอนของเจนเนติกอัลกอริทึมอย่างง่าย*

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

01101

11000

01000

10011

ค่าสตริงของประชากรเริ่มต้นนี้ เกิดจากการสุ่มค่า ทั้งหมด 20 ครั้งหรือ สตริงแต่ละตัวทำการสุ่ม 5 ครั้ง

3.2.2. ประชากรรุ่นเก่า (Old Population)

ประชากรรุ่นเก่า คือสตริงที่จะถูกคัดเลือกไปเป็นต้นแบบสำหรับสร้างประชากรรุ่นใหม่ (New Population) โดยประชากรรุ่นเก่าชุดแรกคือประชากรเริ่มต้นนั่นเอง

3.2.3 การดำเนินการของ SGA

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

- **รีโปรดักชัน** คือกระบวนการที่สตริงแต่ละตัวเลียนแบบค่าฟังก์ชันเป้าหมาย f(x) โดยที่ฟังก์ชันนี้อาจเป็นการวัด ผลตอบแทน ค่าอรรถประโยชน์ หรือสิ่งที่ต้องการให้เป็นค่าสูงสุด หรือค่าความเหมาะสม (Fitness) สตริงที่มีความเหมาะสมสูงกว่าก็จะมีความน่าจะเป็นในการสนับสนุนลูกหลานรุ่นต่อไปสูงด้วย ตัวปฏิบัติการนี้เกิดขึ้นจากกระบวนการคัดเลือกตามธรรมชาติตามทฤษฎีผู้อยู่รอดที่มีความเหมาะสม (Survival of Fittest) ของ ชาล์ส ดาร์วิน ประชากรที่มีความเหมาะสมในธรรมชาติจะมีความสามารถในการรอดพ้นผู้ล่า โรคภัยไข้เจ็บ อุปสรรคอื่นๆที่ต่อต้านการเจริญเติบโตเป็นผู้ใหญ่และสามารถสืบพันธุ์ต่อไปได้ ส่วนฟังก์ชันเป้าหมายจะเป็นสิ่งที่ใช้พิจารณาว่าสตริงที่สร้างขึ้นจะมีชีวิตอยู่หรือตายจากไป

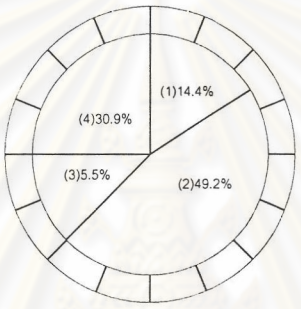
  ตัวปฏิบัติการรีโปรดักชันสามารถสร้างขึ้นได้หลายวิธี วิธีที่ง่ายวิธีหนึ่งคือสร้างจากวงล้อรูลเล็ทที่มีจำนวนช่องเท่ากับจำนวนประชากรสตริง และขนาดของช่องก็เป็นสัดส่วนกับค่าความเหมาะสม ดังรูปที่ 3.3 และค่าความเหมาะสมของฟังก์ชันเป้าหมายของประชากรทั้งสี่แสดงอยู่ในตาราง 3.2

  ค่าความเหมาะสมทั้งหมดโดยรวมจะได้ 1170 และค่ารายละเอียดต่างๆ แสดงดังในตารางที่ 3.3 แสดงถึงวงล้อรูลเล็ทสำหรับการรีโปรดักชัน ซึ่งสร้างจากสัดส่วนของค่า Fitness ของสตริงทั้งหมด เช่นสตริงหมายเลข 1 มีค่าความเหมาะสมเป็น 169 หรือ 14.4% (169/1170) ของค่า Fitness โดยรวมของทั้งประชากร ในการทำการรีโปรดักชันจะหมุนวงล้อเป็นจำนวน 4 ครั้งหรือเท่ากับจำนวนสตริง เช่น สตริงหมายเลข 1 มีค่าเป็น 169 คิดเป็น 14.4% ของค่าความเหมาะสมทั้งหมด ดังนั้นเมื่อหมุนรูลเล็ท 1 ครั้งก็จะมีความน่าจะเป็นที่จะถูกเลือกเท่ากับ 0.144 ในการหมุนรูลเล็ทแต่ละครั้งจะได้ตัวแทนในการสืบพันธุ์ (Reproduction Candidate) สตริงที่มีความเหมาะสมสูงจะถูกคัดเลือกสำหรับการสืบพันธุ์การรีโปรดักชันสำหรับสตริง

ลูกหลานในรุ่นต่อไป เมื่อสตริงมีรูปร่างที่แน่นอนแล้วก็จะถูกส่งไปเข้าเมทติงพูล (Mating Pool) เพื่อที่จะผ่านกระบวนการของตัวปฏิบัติการอื่นต่อไป

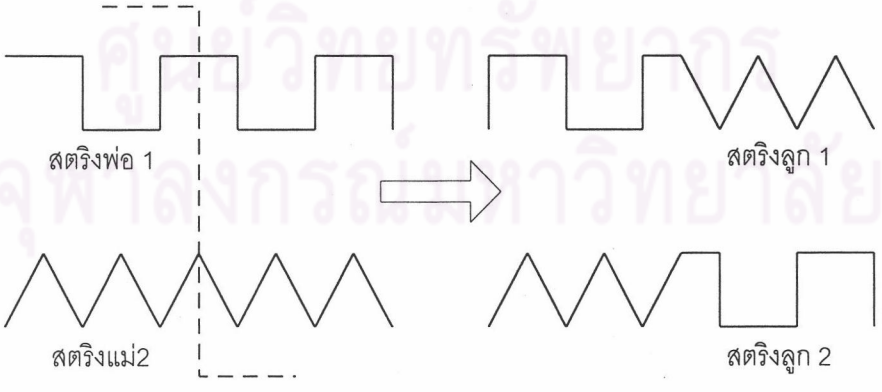
ตารางที่ 3.2 *กลุ่มประชากรตัวอย่างและค่าความเหมาะสม*

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 *การรีโปรดักชันอย่างง่ายด้วยวิธีการใช้วงล้อรูลเล็ทที่มีขนาดของแต่ละช่องเป็นสัดส่วนกับค่าความเหมาะสม*

- **การครอสโอเวอร์** หลังจากประชากรทั้งหมดผ่านกระบวนการรีโปรดักชันแล้ว จะทำการจับคู่สมาชิกในเมทติงพูลหรือกลุ่มประชากรทั้งหมดอย่างสุ่มและทำการไขว้สลับค่าที่อยู่หลังตำแหน่งที่เลือกไว้จากการสุ่มหรือ ทำการแลกเปลี่ยนสวนกัน



รูปที่ 3.4 *การครอสโอเวอร์อย่างง่ายเพื่อให้เห็นถึงการเปลี่ยนแปลงสตริงและการแลกเปลี่ยนข่าวสารโดยเลือกตำแหน่งไขว้แบบสุ่ม*

การเลือกตำแหน่งที่จะทำการครอสโอเวอร์ จะทำโดยการสุ่มค่าที่เป็นจำนวนเต็มตำแหน่งที่ k ช่วงของสตริงที่เลือกจะอยู่ในช่วง [2, t-1] โดยที่ t คือ ตำแหน่งสุดท้ายของสตริงใหม่ทั้งสองก็จะมีการสลับอักขระตั้งแต่ตำแหน่งที่ k+1 จนถึง t ยกตัวอย่างเช่น พิจารณาสตริง A_1, A_2 จากประชากรเริ่มต้น

A_1 = 0110 | 1
A_2 = 1100 | 0

สมมุติว่าเลือกจำนวนสุ่มระหว่าง 1 ถึง 4 และได้ค่า k = 4 (แสดงโดยใช้สัญลักษณ์ " | " แทนการแยก) ผลของการครอสโอเวอร์สตริงที่เป็นประชากรรุ่นใหม่จะมีสัญลักษณ์ " ' "

A'_1 = 0110 | 0
A'_2 = 1100 | 1

- **การมิวเตชัน** มิวเตชันเป็นสิ่งที่จำเป็นถึงแม้ว่ารีโปรดักชันและครอสโอเวอร์ช่วยให้การค้นหาเป็นไปอย่างมีประสิทธิภาพในบางครั้งก็มีการสูญเสียส่วนที่สำคัญไป (ค่า 1 หรือ 0 ในบางตำแหน่ง) การมิวเตชันจะป้องกันส่วนที่สูญเสียที่ไม่อาจเรียกคืนได้ (Irrecovery Loss) ในบางครั้งการหาคำตอบของ SGA คำตอบอาจติดอยู่ใน Local Optimal การมิวเตชันด้วยอัตราส่วนที่เหมาะสมจะทำให้คำตอบสามารถหลุดออกจาก Local Optimal หรืออาจกล่าวได้ว่าโอเปอร์เรเตอร์ของการมีมิวเตชันเป็นการเปลี่ยนแปลงค่าตำแหน่งสตริงแบบสุ่ม จากปัญหาที่พิจารณาค่าจะเปลี่ยนแปลงจาก 0 เป็น 1 หรือ 1 เป็น 0 โดยการเลือกตำแหน่งที่จะทำการมิวเตชันอย่างสุ่ม อัตราการมิวเตชันในธรรมชาติจะมีค่าค่อนข้างต่ำ ในการนำไปใช้งานจะต้องมีการพิจารณาอย่างเหมาะสม

3.2.4 ประชากรรุ่นใหม่ (New population)

สตริงทั้งหมดที่ได้จากกระบวนการของ GAs เรียกว่าประชากรรุ่นใหม่หรือเจนเนอเรชัน (Generation) รุ่นใหม่ซึ่งจะกลายเป็นประชากรรุ่นเก่า สำหรับการดำเนินการต่อไปกระบวนการของ SGA จะทำซ้ำไปเรื่อยๆ จนกว่าจำนวนเจนเนอเรชันจะมากกว่าจำนวนเจนเนอเรชันที่กำหนดไว้สูงสุด

Surech (1995) ได้พิจารณาถึงการหาขนาดของประชากร จากอัตราส่วนของวิธีการที่ทั้งหมดของคำตอบที่เป็นไปไม่ได้และอัตราส่วนของวิธีการที่ทั้งหมดของคำตอบที่เป็นไปได้

$$\lim_{n\to\infty}\frac{n^n}{n!} \approx \lim_{n\to\infty}\frac{(2\pi)^{1/2}\left(\frac{n}{e}\right)^n}{n^n} \qquad (3.1)$$
$$= \lim_{n\to\infty}\frac{(2\pi)^{1/2}}{e^n} = 0$$

จากสมการที่ (3.1) ให้

n คือขนาดของปัญหา

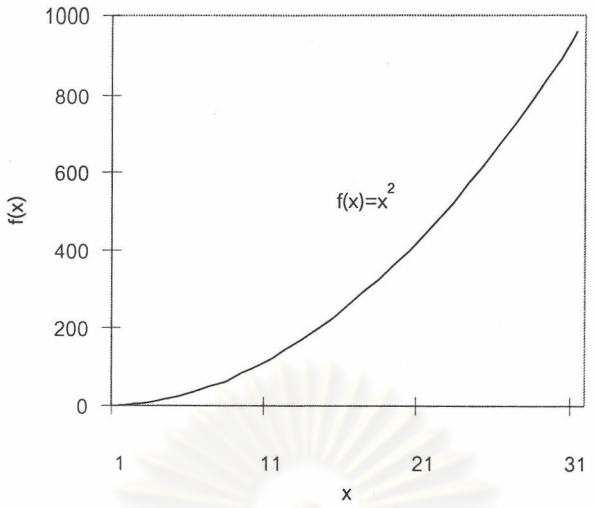
$n!$ คือจำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงที่เป็นไปได้

n^n คือจำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงทั้งหมด

สามารถสรุปได้ว่า ความน่าจะเป็นของในการสร้างประชากรคำตอบอย่างสุ่ม มีค่าเป็นศูนย์เมื่อ n มีค่ามากขึ้น สมมติว่า n มีค่าเป็น 8 ดังนั้น $8! / 8^8 = 1/ 416.1 = 2.403*10^{-3}$ หรืออาจกล่าวได้ว่า โอกาสที่จะได้คำตอบที่ถูกต้องเป็น 1 ใน 416 ของคำตอบที่เป็นไปได้ทั้งหมด ถ้ากำหนดจำนวนประชากรเป็น 100 และทำการคำนวณเพียงแค่ 1 เจนเนอเรชันก็ไม่อาจคาดได้ว่าจะได้คำตอบที่ดี การกำหนดจำนวนประชากรเริ่มต้นและจำนวนเจนเนเรชันทั้งหมดจึงมีผลในการหาคำตอบ

3.3 ตัวอย่างการใช้เจนเนติกอัลกอริทึมในการหาคำตอบของฟังก์ชัน

เนื้อหาในส่วนนี้จะเป็นการประยุกต์ใช้ GAs ในการแก้ปัญหา Optimization หาค่าสูงสุดของฟังก์ชัน $f(x) = x^2$ ทีละขั้นตอน โดย x เป็นตัวแปรที่มีค่าเปลี่ยนแปลงระหว่าง 1 ถึง 31 ดังรูปที่ 3.5 แสดงถึงลักษณะฟังก์ชัน f(x) สำหรับปัญหานี้ตัวแปร x จะถูกเข้ารหัสให้เป็นไบนารี ที่มีความยาวสตริง 5 บิท



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

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

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

ก) การสุ่มสตริงเริ่มต้นและการรีโปรดักชัน

หมายเลขสตริง	ประชากรเริ่มต้น (สร้างขึ้นแบบสุ่ม)	ค่า x (unsigned integer)	$f(x) = x^2$	Pselect $f_i/\sum f$	Expected count $f_i/\bar{f}$	Actual Count (จากวงล้อรูลเล็ท)
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

ตารางที่ 3.3 *การคำนวณหาคำตอบของ SGA กับฟังก์ชัน* $f(x) = x^2$ *(ต่อ)*

ข) *การครอสโอเวอร์*

เมทติงพูลหลังจากการรีโปรดักชัน	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

1. ประชากรเริ่มแรกทั้งสี่ตัว ในแต่ละตัวได้มาจากการสุ่มโยนเหรียญ 5 ครั้ง (มี 5 บิท)
2. รีโปรดักชันได้จากการหมุนวงล้อรูลเล็ท
3. ครอสโอเวอร์ได้จากการโยนเหรียญสองเหรียญแล้วทำการถอดรหัส (TT = 00_2 = 0 = ตำแหน่งที่ไขว้คือ 1, HH = 11_2 = 3 = ตำแหน่งที่ไขว้คือ 4)
4. ความน่าจะเป็นของครอสโอเวอร์กำหนดให้เป็นหนึ่ง $p_c = 1.0$
5. ความน่าจะเป็นของมิวเตชันเป็น 0.001, $p_m = 0.001$, expected mutation = 5*4*0.001 =0.02 ไม่มีค่า expected mutation ระหว่างประชากรเดี่ยว

ประชากรรุ่นต่อไป จะเริ่มต้นกระบวนการรีโปรดักชันจากเมทติงพูล โดยการหมุนวงล้อรูลเล็ท 4 ครั้ง ได้สตริงหมายเลข 1 และ 4 ได้รับการคัดเลือกไปยังรุ่นต่อไป 1 ครั้ง สตริง 2 ได้รับการคัดเลือกไปยังรุ่นต่อไป 2 ครั้ง สตริง 3 ไม่ได้รับการคัดเลือกไปยังรุ่นต่อไปเลย เมื่อเปรียบเทียบจำนวนครั้งที่ถูกคัดเลือกที่คาดหวัง หรือ Expected Count (หาได้จาก $f_i / \bar{f}$) กับจำนวนครั้งที่ถูกคัดเลือกจริงจะเห็นได้ว่ามีค่าใกล้เคียงกัน ค่าที่ดีที่สุดจะมีโอกาสที่จะถูกคัดเลือกมากกว่า ส่วนค่าที่ไม่ดีก็จะตายจากไป

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

ตัวปฏิบัติการสุดท้ายคือมิวเตชันซึ่งจะเปลี่ยนค่าเป็นบิทต่อบิท สมมติความน่าจะเป็นของการมิวเตชันในการทดสอบเป็น 0.001 ตำแหน่งที่จะเปลี่ยนแปลงทั้งหมดมี 20 บิท (ได้จากจำนวนสตริง*จำนวนบิทของสตริงแต่ละตัว 5*4=20) เพราะฉะนั้นตำแหน่งบิทที่จะมิวเตชันของประชากรรุ่นนี้คือ 20 * .001 = 0.02 บิท จากการคำนวณจะเห็นได้ว่าไม่มีบิทใดต้องทำการมิวเตชันสำหรับค่าความน่าจะเป็นนี้ นั่นก็คือไม่มีบิทใดที่จะต้องเปลี่ยนค่าจาก 1 เป็น 0 หรือ 0 เป็น 1 สำหรับประชากรรุ่นนี้ แต่สมมติว่าถ้าตำแหน่งบิทที่จะมิวเตชันของประชากรรุ่นนี้คือ 5 ดังนั้นตำแหน่งบิทที่ 5 จะต้องทำการเปลี่ยนค่าจาก 0 เป็น 1 หรือ 1 เป็น 0

หลังจากผ่านการรีโปรดักชัน ครอสโอเวอร์และมิวเตชัน ประชากรรุ่นใหม่ก็พร้อมที่จะถูกทดสอบ โดยทำการเข้ารหัสสตริงใหม่คำนวณหาค่า x และค่าฟังก์ชัน f(x) ตารางที่ 3.3ข แสดงถึงผลจากการทดลองจะเห็นได้ว่ากระบวนการที่เกี่ยวข้องกับความน่าจะเป็นทำให้ค่าสมรรถนะดีขึ้น ค่าความเหมาะสมของประชากรโดยเฉลี่ยมีค่าเพิ่มขึ้นจาก 293 เป็น 439 ในขณะที่ค่าความเหมาะสมสูงสุดมีค่าเพิ่มขึ้นจาก 576 เป็น 729 ถึงแม้ว่ากระบวนการสุ่มจะช่วยทำให้ค่าต่างๆสูงขึ้นแต่ค่าต่างๆที่เพิ่มขึ้นเหล่านี้ไม่ใช่ความบังเอิญ ค่าสตริงที่ดีที่สุดของประชากรเริ่มแรกคือ (11000) จะมีการเลียนแบบ 2 ครั้งเนื่องจากเป็นค่าที่สูงเกินกว่าค่าเฉลี่ย เมื่อรวมกับค่าสตริงตัวต่อไป (10011) แบบสุ่มและทำการไขว้แบบสุ่มในตำแหน่งที่สองก็จะได้ผลลัพธ์เป็น (11011) ซึ่งก็จะเป็นค่าที่ดีเช่นกัน

ค่าพารามิเตอร์ของ SGA มีความสำคัญอย่างมาก ค่าพารามิเตอร์เหล่านี้ในบางครั้งจำเป็นต้องมีการปรับเปลี่ยนไปตามรูปแบบของปัญหาเพื่อให้ได้คำตอบที่ดี แต่ในบางครั้งก็ไม่อาจที่จะหาคำตอบที่ดีได้เนื่องจาก (Michalewicz, 1992)

1 การเข้ารหัสของปัญหาผิดพลาด ทำให้เจนเนติกอัลกอริทึมหาคำตอบผิดพลาด
2 ขีดจำกัดของจำนวนประชากร ในทางทฤษฎีแล้วมีค่าเป็นอนันต์
3 ขีดจำกัดของจำนวนเจนเนอเรชัน ในทางทฤษฎีแล้วมีค่าเป็นอนันต์

ค่าพารามิเตอร์เหล่านี้ไม่สามารถกำหนดให้เป็นอนันต์ได้ในทางปฏิบัติเนื่องจากข้อจำกัดต่างๆของคอมพิวเตอร์

3.4 สรุปท้ายบท

เจนเนติกอัลกอริทึมเป็นวิธีการค้นหาคำตอบวิธีหนึ่ง โดยมีพื้นฐานจากกระบวนการคัดเลือกทางธรรมชาติ ข้อดีของเจนเนติกอัลกอริทึมเมื่อเปรียบเทียบกับการค้นหาแบบอื่นคือมี

ความคงทนต่อความไม่เที่ยงตรงแม่นยำและความไม่แน่นอนหรือคลุมเครือของปัญหา และสามารถควบคุมได้ โดยมีความน่าเชื่อถือและค่าใช้จ่ายต่ำ

วิธีการค้นหาของเจนเนติกอัลกอริทึมจะแตกต่างกับวิธีการค้นหาและการทำ Optimization แบบอื่นๆ คือ

- เจนเนติกอัลกอริทึมจะใช้งานโดยการเข้ารหัสสตริงเป็นชุดพารามิเตอร์
- เจนเนติกอัลกอริทึมเป็นการค้นหาจากทั้งประชากรไม่ใช่เพียงตำแหน่งเดียว
- เจนเนติกอัลกอริทึมจะใช้ข่าวสารที่เป็นผลลัพธ์ (ฟังก์ชันเป้าหมาย) โดยไม่ใช้อนุพันธ์หรือความรู้อื่น ๆ
- เจนเนติกอัลกอริทึมจะเป็นวิธี Probabilistic ไม่ใช่ Deterministic

โอเปอร์เรเตอร์ต่างๆของเจนเนติกอัลกอริทึมได้แก่

- รีโปรดักชัน คือกระบวนการคัดเลือกสตริงมีความเหมาะสมสูงเพื่อเป็นคำตอบเริ่มต้นให้กับประชากรรุ่นต่อไป โดยอาศัยทฤษฎีของ ชาร์ล ดาร์วินที่ว่า สิ่งมีชีวิตที่แข็งแรงกว่ามีโอกาสอยู่รอดในสภาวะนั้นๆได้มากกว่า
- การครอสโอเวอร์ คือกระบวนการสร้างสตริงลูกหลานใหม่ จากสตริงพ่อแม่
- การมิวเตชันคือ คือกระบวนการที่ช่วยปรับปรุงสตริงให้ดีขึ้นหรือเลวลง โดยการเปลี่ยนแปลงค่าในบางตำแหน่งของสตริง เพื่อให้เกิดสตริงใหม่

พารามิเตอร์ต่างๆของเจนเนติกอัลกอริทึมได้แก่

- จำนวนประชากร
- จำนวนเจนเนอเรชัน
- ค่าความน่าจะเป็นของการครอสโอเวอร์
- ค่าความน่าจะเป็นของการมิวเตชัน

พารามิเตอร์ทั้ง 4 ตัวนี้ จะเป็นพารามิเตอร์พื้นฐานที่จำเป็นต้องเลือกค่าที่เหมาะสมในการนำเจนเนติกอัลกอริทึมไปประยุกต์ใช้งานต่างๆ นอกจากนี้ยังมีพารามิเตอร์อื่นๆที่เกี่ยวข้องกับโอเปอร์เรเตอร์ต่างๆ เช่น วิธีการคัดเลือกสตริงคำตอบ (รีโปรดักชัน) วิธีการครอสโอเวอร์และวิธีการมิวเตชัน ซึ่งจะกล่าวถึงพารามิเตอร์ต่างๆของการนำเจนเนติกอัลกอริทึมไปประยุกต์ใช้สำหรับงานวิจัยนี้ในบทที่ 6

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย