

บทที่ 3

ทฤษฎีเกี่ยวกับเจเนติกอัลกอริทึม

เนื้อหาในบทนี้แบ่งออกเป็น 2 ส่วน ในส่วนแรกกล่าวถึงทฤษฎีเบื้องต้นของเจเนติกอัลกอริทึม (Genetic Algorithms: GAs) และตัวอย่างการคำนวณหาค่าคำตอบของเจเนติกอัลกอริทึมอย่างง่าย เพื่อเป็นพื้นฐานสำหรับการนำเจเนติกอัลกอริทึมไปใช้ในการแก้ปัญหาการทำ Optimization สำหรับส่วนที่สอง จะกล่าวถึงทฤษฎีเกี่ยวกับเจเนติกอัลกอริทึมสำหรับปัญหาที่มีหลายวัตถุประสงค์ (Multi-Objective Genetic Algorithms: MOGA)

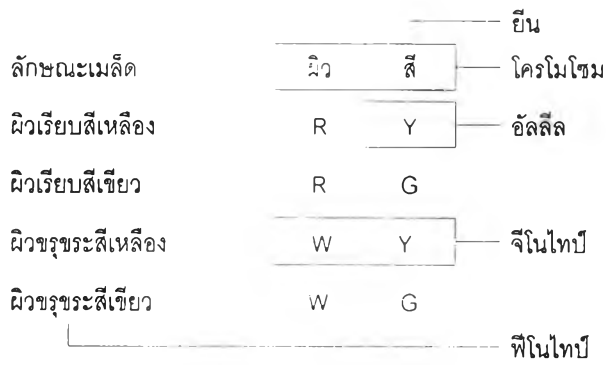
3.1 เจเนติกอัลกอริทึม

ในปัจจุบันการหาคำตอบของปัญหาบางประเภท เช่น ปัญหาการจัดสรรทรัพยากรที่มีอยู่อย่างจำกัดเพื่อให้เกิดประสิทธิภาพสูงสุดและปัญหาในการคำนวณต้นทุนต่ำสุด เป็นต้น สามารถหาคำตอบได้หลายวิธี วิธีการที่ง่ายที่สุดในการหาคำตอบคือวิธีการทางฮิวริสติกต่างๆ ซึ่งอาจได้คำตอบที่ไม่ดีนัก ในปัจจุบันนักวิทยาศาสตร์ได้นำความรู้เกี่ยวกับทฤษฎีหรือกฎเกณฑ์ทางธรรมชาติมาช่วยในการหาคำตอบหรือศึกษาวิจัย โดยมีเป้าหมายหลักในการใช้ประโยชน์ของความคงทน (Robustness) ต่อความไม่เที่ยงตรงแม่นยำ (Accuracy) ความไม่แน่นอน (Uncertainty) หรือความคลุมเครือของปัญหา (Vague) หลักการเหล่านี้สามารถพบได้จากวิธีการต่างๆ เช่น ระบบโครงข่ายประสาทเทียม (Neural Network) ฟัซซีลอจิก (Fuzzy Logic) และ GAs (Goldberg, 1989) ปัญหาที่พบส่วนใหญ่เป็นปัญหาที่ไม่เที่ยงตรงและคลุมเครือ ซึ่งหาต้องการคำตอบที่เที่ยงตรงและมีความแน่นอนสูงมากก็ย่อมมีค่าใช้จ่ายที่สูงมาก ดังนั้นวิธีการที่สามารถแก้ปัญหาที่คลุมเครือโดยได้คำตอบที่ใกล้เคียงสามารถยอมรับได้ ใช้เวลาในการหาคำตอบไม่มากนัก และมีค่าใช้จ่ายพอประมาณ ย่อมดีกว่าวิธีที่ได้ความเที่ยงตรงสูงแต่มีค่าใช้จ่ายที่สูง วิธีการหาคำตอบที่ได้อย่างหนึ่งได้แก่วิธีการของ GAs โดยอาศัยทฤษฎีในการถ่ายทอดลักษณะต่างๆทางกรรมพันธุ์ไปสู่ยังลูกหลาน ซึ่งสามารถนำมาพัฒนาใช้ในการหาคำตอบที่ต้องการได้

3.1.1 พันธุศาสตร์กับเจเนติกอัลกอริทึม

Mendel บิดาแห่งวิชาพันธุศาสตร์ ค้นพบว่าลักษณะต่างๆของสิ่งมีชีวิต เช่น ลักษณะผิวของเมล็ดพืช สีของเมล็ดพืช ฯลฯ ที่ถูกถ่ายทอดไปยังลูกหลานนั้นถูกควบคุมโดยหน่วยควบคุมลักษณะที่เรียกว่ายีน (Gene) และลักษณะย่อยของยีนเรียกว่าอัลลีล (Allele) เช่นยีน ควบคุมลักษณะผิวของเมล็ดจะมีอัลลีลเป็นผิวเรียบและผิวขรุขระ

เป็นต้น ซึ่งแต่ละยีนจะเรียงตัวอยู่บนโครโมโซม (Chromosome) ภายในเซลล์ ตำแหน่งของยีนแต่ละยีนบนโครโมโซมเรียกว่า โลกัส (Locus) และแต่ละแบบของชุดยีนเรียกว่า จีโนไทป์ (Genotype) ซึ่งแสดงลักษณะภายนอกที่ปรากฏ ซึ่งเรียกว่า ฟีนไทป์ (Phenotype) ดังรูปที่ 3.1ก



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

อักขระ	สตริง		ค่าอักขระ		โครงสร้าง	ค่าพหามิตอร์	ปัญหา
	บิต 1	บิต 2					
	0	0	0	0			x^2
	0	1	1	1			
	1	0	2	4			
	1	1	4	16			

คำตอบของปัญหาซึ่งเป็นค่าของฟังก์ชันวัตถุประสงค์

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

รูปที่ 3.1 เปรียบเทียบลักษณะระหว่างเจเนติกอัลกอริทึมกับลักษณะทางพันธุศาสตร์

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)
ฟีนไทป์ (Phenotype)	โครงสร้างคำตอบ (A decode structure)

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

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

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

3.1.2 ความหมายของเจเนติกอัลกอริทึม

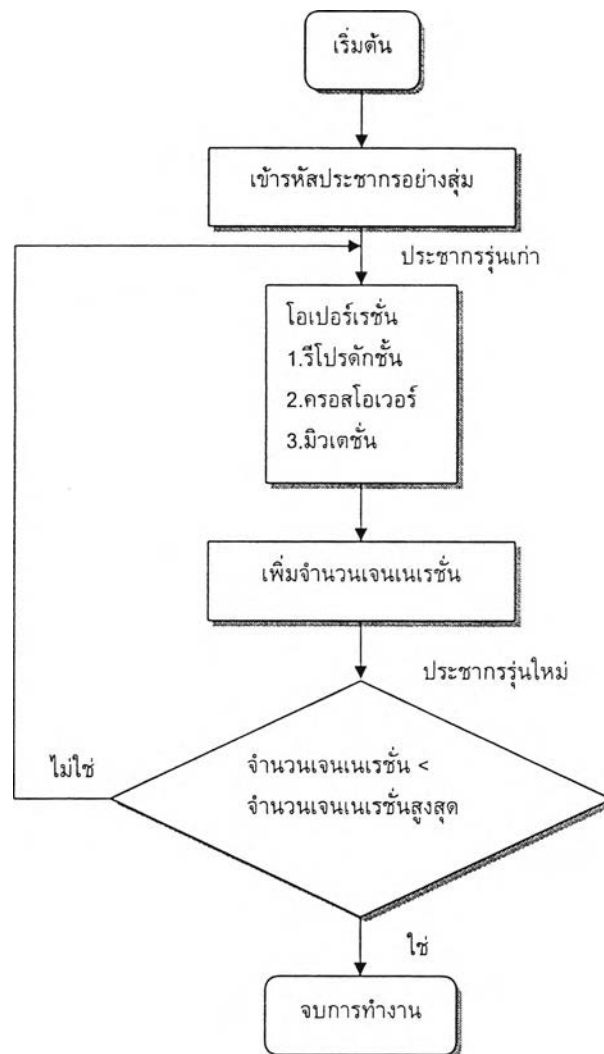
GAs เป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานมาจากกระบวนการคัดเลือกทางธรรมชาติ (Natural Selection) และ กระบวนการคัดเลือกทางพันธุศาสตร์ (Natural Genetics Selection) โดยการคัดเลือกสตริง (String) ที่มีความเหมาะสมจากกลุ่มของสตริงทั้งหมดด้วยวิธีการสุ่ม จากการนำสตริงเหล่านี้ไปผ่านกระบวนการคัดเลือกสตริงที่มีความเหมาะสม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

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

- GAs ทำงานโดยการเข้ารหัสสตริงเป็นชุดพารามิเตอร์
- GAs เป็นการค้นหาจากทั้งประชากรไม่ใช่ค้นหาจากเพียงตำแหน่ง ๆ เดียว

- GAs ใช้ข่าวสารที่เป็นผลลัพธ์ (ฟังก์ชันเป้าหมาย) โดยไม่ใช้การอนุพันธ์หรือความรู้อื่นๆ
- GAs จะเป็นวิธี Probabilistic ไม่ใช่ Deterministic

3.2 เจเนติกอัลกอริทึมอย่างง่าย (Simple Genetic Algorithms)



รูปที่ 3.2 ขั้นตอนของ GAs อย่างง่าย

ขั้นตอนการทำ GAs อย่างง่าย (Simple Genetic Algorithms:SGA) ดังรูปที่ 3.2 มีรายละเอียดดังต่อไปนี้

3.2.1 การเข้ารหัสและสร้างประชากรเริ่มต้นอย่างสุ่ม

ขั้นตอนแรกของ GAs คือ การเข้ารหัสหรือแปลงค่าพารามิเตอร์ให้อยู่ในรูปของสตริงที่มีความยาวแน่นอน ซึ่งวิธีการเข้ารหัสขึ้นอยู่กับรูปแบบของปัญหาแต่ละปัญหา 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 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 รีโพรดักชัน (Reproduction) การครอสโอเวอร์ (Crossover) และการมิวเตชัน (Mutation) ดังมีรายละเอียดดังต่อไปนี้

- **รีโพรดักชัน** คือกระบวนการที่สตริงแต่ละตัวเลียนแบบค่าฟังก์ชันเป้าหมาย $f(x)$ โดยที่ฟังก์ชันนี้อาจเป็นการวัด ผลตอบแทน ค่ารรถประโยชน์ หรือสิ่งที่ต้องการให้เป็นค่าสูงสุด หรือค่าความเหมาะสม (Fitness) สตริงที่มีความเหมาะสมสูงกว่าก็จะมีแนวโน้มจะเป็นในการสนับสนุนลูกหลานรุ่นต่อไปสูงด้วย ตัวปฏิบัติการนี้เกิดขึ้นจากกระบวนการคัดเลือกตามธรรมชาติตามทฤษฎีผู้รอดที่มีความเหมาะสม (Survival of Fittest) ของ ชาลส์ ดาร์วิน ประชากรที่มีความเหมาะสมในธรรมชาติจะ

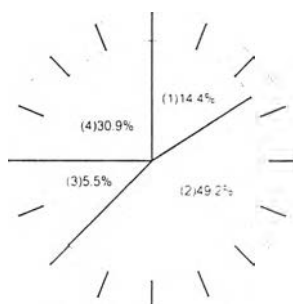
มีความสามารถในการรอดพ้นผู้ล่า โรคภัยไข้เจ็บ อุปสรรคอื่น ๆ ที่ต่อต้านการเจริญเติบโตเป็นผู้ใหญ่ และสามารถสืบพันธุ์ต่อไปได้ ส่วนฟังก์ชันเป้าหมายจะเป็นสิ่งที่ใช้พิจารณาว่าสตริงที่สร้างขึ้นจะมีชีวิตอยู่หรือตายจากไป

ตัวปฏิบัติการรีโพรดักชันสามารถสร้างขึ้นได้หลายวิธี วิธีที่ง่ายวิธีหนึ่งคือสร้างจากวงล้อรูเล็ตที่มีจำนวนช่องเท่ากับจำนวนประชากรสตริง และขนาดของช่องก็เป็นสัดส่วนกับค่าความเหมาะสม ดังรูปที่ 3.3 และค่าความเหมาะสมของฟังก์ชันเป้าหมายของประชากรทั้งสิ้นแสดงอยู่ในตาราง 3.2

ค่าความเหมาะสมทั้งหมดโดยรวมจะได้ 1170 และค่ารายละเอียดต่างๆ แสดงดังในตารางที่ 3.3 แสดงถึงวงล้อรูเล็ตสำหรับการรีโพรดักชัน ซึ่งสร้างจากสัดส่วนของค่าฟิตเนสของสตริงทั้งหมด เช่นสตริงหมายเลข 1 มีค่าความเหมาะสมเป็น 169 หรือ 14.4% (169/1170) ของค่าฟิตเนสโดยรวมของทั้งประชากร ในการทำการรีโพรดักชันจะหมุนวงล้อเป็นจำนวน 4 ครั้งหรือเท่ากับจำนวนสตริง เช่นสตริงหมายเลข 1 มีค่าเป็น 169 คิดเป็น 14.4% ของค่าความเหมาะสมทั้งหมด ดังนั้นเมื่อหมุนรูเล็ต 1 ครั้งก็มีความน่าจะเป็นที่จะถูกเลือกเท่ากับ 0.144 ในการหมุนรูเล็ตแต่ละครั้งจะได้ตัวแทนในการสืบพันธุ์ (Reproduction Candidate) สตริงที่มีความเหมาะสมสูงจะถูกคัดเลือกสำหรับการสืบพันธุ์การรีโพรดักชันสำหรับสตริงลูกหลานในรุ่นต่อไป เมื่อสตริงมีรูปร่างที่แน่นอนแล้วก็จะถูกส่งไปเข้าเมทติ้งพูล (Mating Pool) เพื่อที่จะผ่านกระบวนการของตัวปฏิบัติการอื่นต่อไป

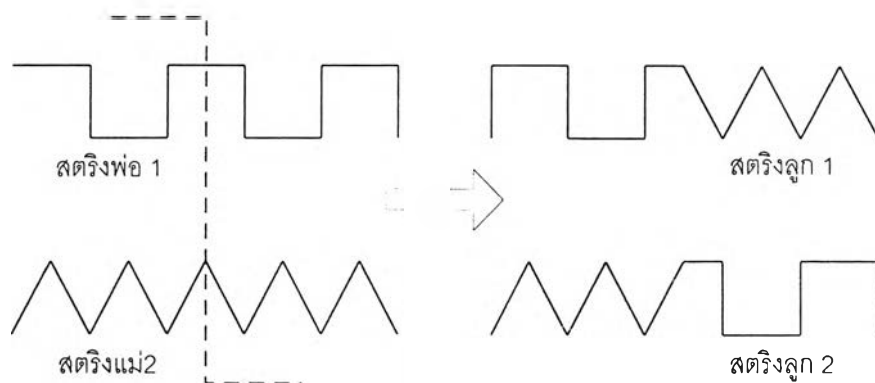
ตาราง 3.2 กลุ่มประชากรตัวอย่างและค่าความเหมาะสม

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

- การครอสโอเวอร์ หลังจากประชากรทั้งหมดผ่านกระบวนการรีโพรดักชันแล้ว จะทำการจับคู่สมาชิกในเมตาดิงพูลหรือกลุ่มประชากรทั้งหมดอย่างสุ่มและทำการไขว้สลับค่าที่อยู่หลังตำแหน่งที่เลือกไว้จากการสุ่มหรือ ทำการแลกเปลี่ยนส่วนกัน



รูปที่ 3.4 การครอสโอเวอร์อย่างง่ายเพื่อให้เห็นถึงการเปลี่ยนแปลงสตริงและการแลกเปลี่ยนข่าวสารโดยเลือกตำแหน่งไขว้แบบสุ่ม

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

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

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

$$A'_1 = 01100$$

$$A'_2 = 11001$$

- การมิวเตชัน มิวเตชันเป็นสิ่งที่จำเป็นถึงแม้ว่ารีโพรดักชันและครอสโอเวอร์ช่วยให้การค้นหาเป็นไปอย่างมีประสิทธิภาพในบางครั้งก็มีการสูญเสียส่วนที่สำคัญไป (ค่า 1 หรือ 0 ในบางตำแหน่ง) การมิวเตชันจะป้องกันส่วนที่สูญเสียที่ไม่อาจเรียกคืนได้ (Irrecovery Loss) ในบางครั้งการหาคำตอบของ SGA คำตอบอาจติดอยู่ใน Local Optima การมิวเตชันด้วยอัตราส่วนที่เหมาะสมจะทำให้คำตอบสามารถหลุดออกจาก Local Optima หรืออาจกล่าวได้ว่าโอเปอร์เรเตอร์ของการมีมิวเตชันเป็นการเปลี่ยนแปลงค่าตำแหน่งสตริงแบบสุ่ม จากปัญหาที่พิจารณาค่าจะเปลี่ยนแปลงจาก 0 เป็น 1 หรือ 1 เป็น 0 โดยการเลือกตำแหน่งที่จะทำการมิวเตชันอย่างสุ่ม อัตรา

การมีเวกชันในธรรมชาติจะมีค่าค่อนข้างต่ำ ในการนำไปใช้งานจะต้องมีการพิจารณาอย่างเหมาะสม

3.2.4 ประชากรรุ่นใหม่ (New population)

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

Surech (1995) ได้พิจารณาถึงการหาขนาดของประชากร จากอัตราส่วนของวิธีการที่ทั้งหมดของคำตอบที่เป็นไปไม่ได้และอัตราส่วนของวิธีการที่ทั้งหมดของคำตอบที่เป็นไปได้

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

จากสมการที่ (3.1) ให้

n คือขนาดของปัญหา

$n!$ คือจำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงที่เป็นไปได้

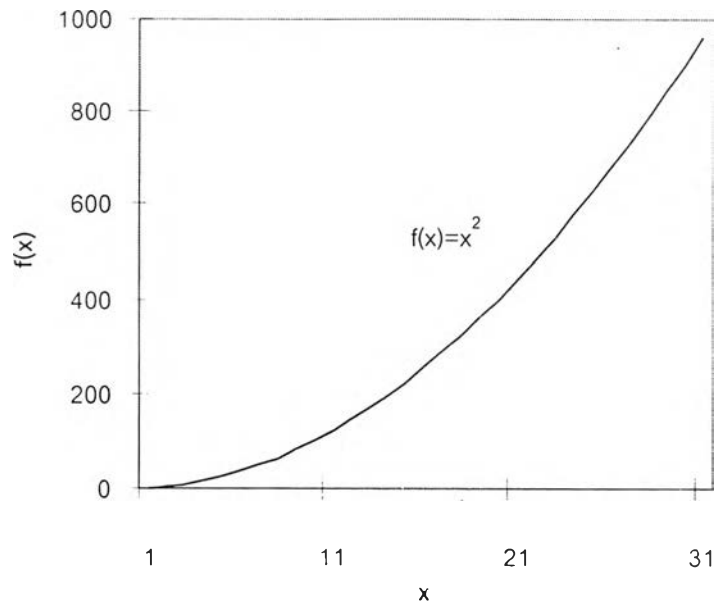
n^n คือจำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงทั้งหมด

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 ตัวอย่างการใช้เจเนติกอัลกอริทึมในการหาคำตอบของฟังก์ชัน

เนื้อหาในส่วนนี้จะเป็นการประยุกต์ใช้ GAs ในการแก้ปัญหา Optimization หาค่าสูงสุดของฟังก์ชัน $f(x) = x^2$ ที่ละชั้นตอน โดย x เป็นตัวแปรที่มีค่าเปลี่ยนแปลงระหว่าง 1 ถึง 31 ดังรูป

ที่ 3.5 แสดงถึงลักษณะฟังก์ชัน $f(x)$ สำหรับปัญหานี้ตัวแปร x จะถูกเข้ารหัสให้เป็นไบนารี ที่มีความยาวสตริง 5 บิต



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

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

ตารางที่ 3.3 การคำนวณหาค่าตอบของ 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 (จากวงล้อรูล เล็ต)
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

ข) การครอสโอเวอร์

เมทติ้งพูลหลังจากการรีโปรดักชัน	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 บิต)
2. รีโปรดักชันได้จากการหมุนวงล้อรูเล็ต
3. ครอสโอเวอร์ได้จากการโยนเหรียญสองเหรียญแล้วทำการถอดรหัส ($TT = 00_2 = 0 =$ ตำแหน่งที่ไขว้คือ 1, $HH = 11_2 = 3 =$ ตำแหน่งที่ไขว้คือ 4)
4. ความน่าจะเป็นของครอสโอเวอร์กำหนดให้เป็นหนึ่ง $p_c = 1.0$
5. ความน่าจะเป็นของมิวเตชันเป็น 0.001, $p = 0.001$, expected mutation = $5 \cdot 4 \cdot 0.001 = 0.2$ ไม่มีค่า expected mutation ระหว่างประชากรเดียว

ประชากรรุ่นต่อไป จะเริ่มต้นกระบวนการรีโปรดักชันจากเมทติ้งพูล 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 3.3x อีกครั้ง จะเห็นได้ว่าจากการสุ่มจับคู่ในเมทติ้งพูล สตริงหมายเลข 2 จะจับคู่กับสตริงหมายเลข 1 และมีตำแหน่งการไขว้คือ 4 สตริงทั้งสองคือ 01101 และ 11000 เมื่อทำการไขว้จะได้สตริงตัวใหม่คือ 01100 และ 11001 สตริงที่เหลือในเมทติ้งพูลจะทำการไขว้กันในตำแหน่งที่สองดังแสดงในตารางที่ 3.3 ข

ตัวปฏิบัติการสุดท้ายคือมิวเตชันซึ่งจะเปลี่ยนค่าเป็นบิตต่อบิต สมมุติความน่าจะเป็นของการมิวเตชันในการทดสอบเป็น 0.001 ตำแหน่งที่จะเปลี่ยนแปลงทั้งหมดมี 20 บิต (ได้จากจำนวนสตริง*จำนวนบิตของสตริงแต่ละตัว $5 \cdot 4 = 20$) เพราะฉะนั้นตำแหน่งบิตที่จะมิวเตชันของ

ประชากรรุ่นนี้คือ $20 * .001 = 0.02$ บิท จากการคำนวณจะเห็นได้ว่าไม่มีบิทใดต้องทำการมิวเตชันสำหรับค่าความน่าจะเป็นนี้ นั่นก็คือไม่มีบิทที่จะต้องเปลี่ยนค่าจาก 1 เป็น 0 หรือ 0 เป็น 1 สำหรับประชากรรุ่นนี้ แต่สมมติว่าถ้าตำแหน่งบิทที่จะมิวเตชันของประชากรรุ่นนี้คือ 5 ดังนั้นตำแหน่งบิทที่ 5 จะต้องทำการเปลี่ยนค่าจาก 0 เป็น 1 หรือ 1 เป็น 0

หลังจากผ่านการวิโปรดักชัน คrossover และมิวเตชัน ประชากรรุ่นใหม่ก็พร้อมที่จะถูกทดสอบ โดยทำการเข้ารหัสสตริงใหม่คำนวณหาค่า x และค่าฟังก์ชัน $f(x)$ ตารางที่ 3.3 ข. แสดงถึงผลจากการทดลองจะเห็นได้ว่ากระบวนการที่เกี่ยวข้องกับความน่าจะเป็นจะทำให้ค่าสมรรถนะดีขึ้น ค่าความเหมาะสมของประชากรโดยเฉลี่ยมีค่าเพิ่มขึ้นจาก 293 เป็น 439 ในขณะที่ค่าความเหมาะสมสูงสุดมีค่าเพิ่มขึ้นจาก 576 เป็น 729 ถึงแม้ว่ากระบวนการสุ่มจะช่วยให้ค่าต่างๆ สูงขึ้นแต่ค่าต่างๆ ที่เพิ่มขึ้นเหล่านี้ไม่ใช่ความบังเอิญ ค่าสตริงที่ดีที่สุดของประชากรเริ่มแรกคือ (11000) จะมีการเลียนแบบ 2 ครั้งเนื่องจากเป็นค่าที่สูงเกินกว่าค่าเฉลี่ย เมื่อรวมกับค่าสตริงตัวต่อไป (10011) แบบสุ่มและทำการไขว้แบบสุ่มในตำแหน่งที่สองก็จะได้ผลลัพธ์เป็น (11011) ซึ่งก็จะเป็นค่าที่ดีเช่นกัน

ค่าพารามิเตอร์ของ SGA มีความสำคัญอย่างมาก ค่าพารามิเตอร์เหล่านี้ในบางครั้งจำเป็นต้องมีการปรับเปลี่ยนไปตามรูปแบบของปัญหาเพื่อให้ได้คำตอบที่ดี แต่ในบางครั้งก็ไม่อาจที่จะหาคำตอบที่ดีได้เนื่องจาก (Michalewicz, 1992)

- 1 การเข้ารหัสของปัญหาผิดพลาด ทำให้ GAs หาคำตอบผิดพลาด
- 2 ขีดจำกัดของจำนวนประชากร ในทางทฤษฎีแล้วมีค่าเป็นอนันต์
- 3 ขีดจำกัดของจำนวนเจนเนอเรชัน ในทางทฤษฎีแล้วมีค่าเป็นอนันต์

ค่าพารามิเตอร์เหล่านี้ไม่สามารถกำหนดให้เป็นอนันต์ได้ในทางปฏิบัติเนื่องจากข้อจำกัดต่างๆ ของคอมพิวเตอร์

3.4 เจเนติกอัลกอริทึมแบบหลายวัตถุประสงค์

ในอดีตได้มีการคิดค้นวิธีการเพื่อใช้หาคำตอบของปัญหาการหาค่าที่ดีที่สุดทางวิศวกรรมเป็นจำนวนมาก แต่ปัญหาที่ศึกษาส่วนใหญ่มักเป็นปัญหาในอุดมคติ วิธีการส่วนมากมักพิจารณาวัตถุประสงค์ในการแก้ปัญหาเพียงอย่างเดียว แต่อย่างไรก็ตามในความเป็นจริงแล้วปัญหาที่พบส่วนใหญ่มักเป็นปัญหาที่มีหลายวัตถุประสงค์ ดังนั้นจึงต้องอาศัยประสบการณ์ของผู้ออกแบบเป็นหลักในการตัดสินใจเลือกคำตอบที่ดีที่สุด

ในช่วงหลายปีที่ผ่านมา ได้มีการคิดค้นพัฒนาเทคนิคทางคณิตศาสตร์ใหม่ขึ้นมาเพื่อใช้กับปัญหาที่มีหลายวัตถุประสงค์ ส่วนมากมักเป็นการหาคำตอบที่ดีที่สุดออกมาเพียงคำตอบเดียวมากกว่าการหาคำตอบที่เป็นไปได้หลายๆคำตอบแล้วจึงให้ผู้ออกแบบตัดสินใจเลือกภายหลัง วิธีการหนึ่งที่ได้รับ ความสนใจอย่างมากคือวิธีการของเจเนติกอัลกอริทึม ซึ่งงานวิจัยหลาย

ชั้นได้เสนอแนวทางการพัฒนาวิธีการของเจเนติกอัลกอริทึมอย่างง่ายเพื่อให้สามารถนำไปใช้จัดการกับปัญหาแบบหลายวัตถุประสงค์ได้

วิธีการเจเนติกอัลกอริทึมสำหรับปัญหาแบบหลายวัตถุประสงค์ที่มีการพัฒนาขึ้นมา นั้นมีทั้งแบบที่ให้คำตอบที่ดีที่สุดค่าเดียว และแบบที่ให้หลายคำตอบที่เป็นไปได้ ทั้งนี้ขึ้นกับความเหมาะสมและความต้องการของผู้ออกแบบ แต่อย่างไรก็ตาม ถึงแม้จะมีการพัฒนาเจเนติกอัลกอริทึมเพื่อจัดการกับปัญหาหลายวัตถุประสงค์มากมายหลายวิธี แต่วิธีการที่พัฒนาขึ้นนี้ไม่ได้ถูกกำหนดว่าวิธีใดเป็นวิธีที่เหมาะสมที่สุด เนื่องจากเป็นยังเป็นเพียงการพัฒนาในขั้นต้นเท่านั้น

การพัฒนาส่วนมาก เป็นการพัฒนาในส่วนของการคัดเลือกคำตอบที่มีความเหมาะสมที่สุด ในขั้นตอนของการประเมินค่า (Evaluate) และคัดเลือกคำตอบ (Selection) วิธีการที่สามารถแบ่งออกเป็น 4 แบบ (Coello และ Christiansen) ได้แก่

1. Use of Aggregation Function ได้แก่
 - วิธี Weight Sum Approach โดย Jakob et al.
 - วิธี Reduction to Single Objective โดย Ritzel และ Wayland
 - วิธี Goal Attainment โดย Wilson และ Macleod
 - Use of Penalty Function โดย Adehi และ Cheng
2. Non-Pareto Approach ได้แก่
 - วิธี VEGA โดย David Schaffer
 - วิธี Lexicographic Ordering โดย Fourman
 - วิธี Evolutionary Strategies โดย Kursawe
 - วิธี Weight Sum โดย Hajela และ Lin
3. Pareto-based Approach ได้แก่
 - วิธี Pareto-based Assignment โดย Goldberg
 - วิธี Multiple Objective Genetic Algorithm โดย Fonseca และ Fleming
 - วิธี Non-dominated Sorting Genetic Algorithm โดย Srinivas และ Deb
 - วิธี Niche Pareto GA โดย Horn และ Nafpliotis
4. Min-Max Strategy ได้แก่
 - วิธี Weighted Min-Max Strategy โดย Coello และ Christiansen
 - วิธี Min-Max Selection with Sharing โดย Coello และ Christiansen

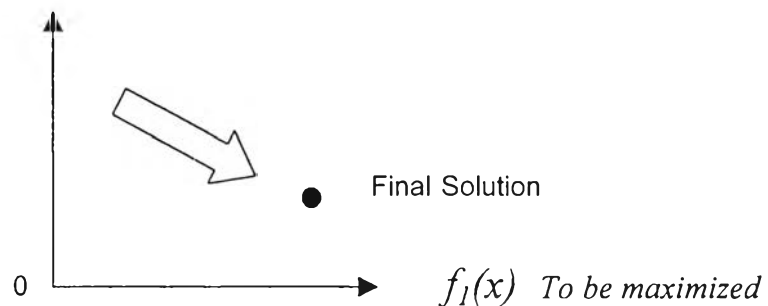
วิธีการทั้ง 4 นี้ วิธีการรวมค่าฟังก์ชันเป็นวิธีที่ง่ายที่สุด วิธีนี้อาศัยหลักการรวมฟังก์ชันวัตถุประสงค์หลายๆฟังก์ชันเข้าเป็นฟังก์ชันเดียวกัน การรวมค่าฟังก์ชันทำได้หลายแบบ แต่วิธีที่เข้าใจได้ง่ายที่สุดคือวิธีการรวมฟังก์ชันโดยอาศัยการให้น้ำหนัก (Weight Sum Approach)

วิธีการรวมฟังก์ชันโดยอาศัยการให้น้ำหนักนี้เป็นวิธีการที่ไม่ยุ่งยากและง่ายต่อการนำไปใช้ วิธีนี้อาศัยการให้น้ำหนักกับวัตถุประสงค์แต่ละอย่าง โดยที่น้ำหนักนี้อาจจะได้มาจากการประมาณความสำคัญของวัตถุประสงค์นั้นๆ แล้วนำเอาน้ำหนักและฟังก์ชันวัตถุประสงค์ทั้งหมดมารวมกันเป็นสมการเส้นตรง (Scalar Fitness Function) โดยมีน้ำหนักของวัตถุประสงค์เป็นสัมประสิทธิ์ของฟังก์ชันวัตถุประสงค์นั้นๆ ดังสมการที่ 3.2

$$f(x) = W_1 f_1(x) + W_2 f_2(x) + W_3 f_3(x) + \dots + W_n f_n(x) \quad (3.2)$$

จากสมการดังกล่าว ถ้ากำหนดน้ำหนักให้กับแต่ละวัตถุประสงค์เป็นค่าเฉพาะหนึ่งๆ จะเห็นได้ว่าทิศทางการหาคำตอบจะมุ่งสู่จุดใดจุดหนึ่งเพียงจุดเดียวเท่านั้น ดังแสดงในรูปที่ 3.6 ดังนั้นวิธีนี้จะให้คำตอบที่ดีที่สุด (Trade-off) สำหรับการกำหนดค่าน้ำหนักแบบหนึ่งๆ เพียงคำตอบเดียว จึงไม่ต้องพึงพาการตัดสินใจของผู้ออกแบบอีกครั้งหนึ่ง

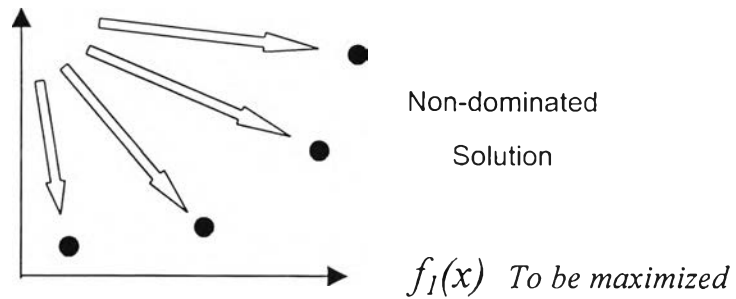
$f_2(x)$ To be minimized



รูปที่ 3.6 ทิศทางการหาคำตอบโดยเจเนติกอัลกอริทึมร่วมกับการรวมฟังก์ชันวัตถุประสงค์แบบกำหนดน้ำหนักเฉพาะ

ปัญหาประการหนึ่งของการกำหนดน้ำหนักเป็นค่าเฉพาะหนึ่งๆนั้นก็คือ ถ้าหากไม่มีข้อมูลเพียงพอก็จะเกิดความยุ่งยากในการกำหนดน้ำหนักของแต่ละวัตถุประสงค์ ในกรณีนี้อาจใช้วิธีการสร้างชุดของคำตอบที่ดี (Non-dominated Solution) แทนโดยการเปลี่ยนค่าน้ำหนักของวัตถุประสงค์ไปเรื่อยๆ เพื่อหาขอบเขตของคำตอบที่ดี แล้วค่อยให้ผู้ออกแบบตัดสินใจเลือก ทิศทางการหาคำตอบโดยการเปลี่ยนค่าน้ำหนักเป็นค่าต่างๆแสดงดังรูปที่ 3.7

$f_2(x)$ To be minimized



รูปที่ 3.7 ทิศทางการหาคำตอบโดยเจเนติกอัลกอริทึมร่วมกับการรวมฟังก์ชันวัตถุประสงค์แบบกำหนดน้ำหนักหลายค่า

ในกรณีที่วัตถุประสงค์ที่พิจารณาขัดแย้งกันหรือมีหน่วยต่างกัน จะไม่สามารถรวมฟังก์ชันวัตถุประสงค์เข้าด้วยกันโดยตรง แต่ต้องทำการเปลี่ยนรูปฟังก์ชันวัตถุประสงค์ทั้งหมดให้สอดคล้องและมีหน่วยพื้นฐานเดียวกันเสียก่อน

3.5 สรุปท้ายบท

GAs เป็นวิธีการค้นหาคำตอบวิธีหนึ่ง โดยมีพื้นฐานจากกระบวนการคัดเลือกทางธรรมชาติ ข้อดีของ GAs เมื่อเปรียบเทียบกับการค้นหาแบบอื่นคือมีความคงทนต่อความไม่เที่ยงตรง แม่นยำและความไม่แน่นอนหรือคลุมเครือของปัญหา และสามารถควบคุมได้ โดยมีความน่าเชื่อถือและค่าใช้จ่ายต่ำ

วิธีการค้นหาของ GAs จะแตกต่างกับวิธีการค้นหาและการทำ Optimization แบบอื่นๆ คือ

- GAs จะใช้งานโดยการเข้ารหัสสตริงเป็นชุดพารามิเตอร์
- GAs เป็นการค้นหาจากทั้งประชากรไม่ใช่เพียงตำแหน่งๆเดียว
- GAs จะใช้ข่าวสารที่เป็นผลลัพธ์ (ฟังก์ชันเป้าหมาย) โดยไม่ใช่อนุพันธ์หรือความรู้อื่น ๆ
- GAs จะเป็นวิธี Probabilistic ไม่ใช่ Deterministic

โอเปอเรเตอร์ต่างๆของ GAs ได้แก่

- รีโปรดั๊กชัน คือกระบวนการคัดเลือกสตรีงที่มีความเหมาะสมสูงเพื่อเป็นคำตอบเริ่มต้นให้กับประชากรรุ่นต่อไป โดยอาศัยทฤษฎีของ ชาร์ล ดาร์วินที่ว่า สิ่งมีชีวิตที่แข็งแรงกว่ามีโอกาสอยู่รอดในสภาวะนั้นๆ ได้มากกว่า
- การครอสโอเวอร์ คือกระบวนการสร้างสตรีงลูกหลานใหม่ จากสตรีงพ่อแม่
- การมิวเตชันคือ คือ กระบวนการที่ช่วยปรับปรุงสตรีงให้ดีขึ้นหรือเลวลง โดยการเปลี่ยนแปลงค่าในบางตำแหน่งของสตรีง เพื่อให้เกิดสตรีงใหม่

พารามิเตอร์ต่างๆของ GAs ได้แก่

- จำนวนประชากร
- จำนวนเจนเนอเรชัน
- ค่าความน่าจะเป็นของการครอสโอเวอร์
- ค่าความน่าจะเป็นของการมิวเตชัน

นอกจากนี้ ในปัจจุบันได้มีการพัฒนาเจเนติกอัลกอริทึมเพื่อใช้กับปัญหาที่มีหลายวัตถุประสงค์ วิธีการของเจเนติกอัลกอริทึมสำหรับปัญหาหลายวัตถุประสงค์นี้จะแตกต่างกับวิธีการของเจเนติกอัลกอริทึมอย่างง่ายในขั้นตอนของการประเมินค่าความเหมาะสม และการคัดเลือกคำตอบต่างๆ

วิธีการรวมฟังก์ชันโดยอาศัยการให้น้ำหนักกับวัตถุประสงค์ก็เป็นวิธีการหนึ่งที่ไม่ยุ่งยากและง่ายต่อการนำไปใช้ วิธีนี้อาศัยการให้น้ำหนักกับแต่ละวัตถุประสงค์ แล้วเอาฟังก์ชันวัตถุประสงค์ทั้งหมดมารวมกันในรูปสมการเส้นตรงที่มีน้ำหนักวัตถุประสงค์เป็นค่าสัมประสิทธิ์ น้ำหนักที่ให้สามารถกำหนดเป็นค่าเฉพาะเพื่อหาคำตอบที่ดีที่สุดค่าเดียว หรืออาจกำหนดหลายค่าเพื่อหาพื้นที่ผิวของคำตอบที่ดีที่สุดแล้วค่อยให้ผู้ตัดสินใจเลือกอีกครั้ง