บทที่ 3

เจนเนติกอัลกอริทึม และการแก้ปัญหาหลายวัตถุประสงค์

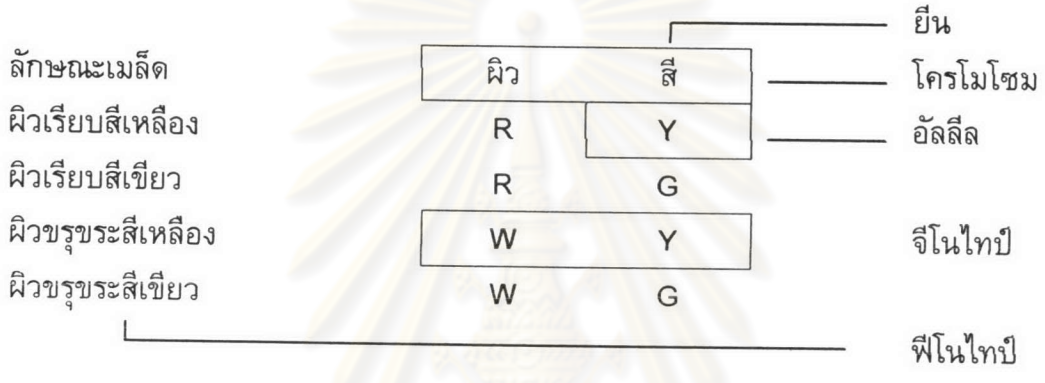
เนื้อหาในบทนี้จะกล่าวถึงทฤษฎีเบื้องต้นของเจนเนติกอัลกอริทึม (Genetic Algorithms หรือ GAs) กับตัวอย่างการคำนวณหาค่าคำตอบของเจนเนติกอัลกอริทึมอย่างง่าย (Simple Genetic Algorithms) เพื่อเป็นพื้นฐานสำหรับการนำเจนเนติกอัลกอริทึมไปใช้ในการแก้ปัญหาการทำ Optimization และตัวอย่างการคำนวณหาคำตอบของเจนเนติกอัลกอริทึม เพื่อนำไปสู่การนำเจนเนติกอัลกอริทึมไปประยุกต์ใช้ในการแก้ปัญหาการจัดผังโรงงาน รวมถึงทฤษฎีเกี่ยวกับเจนเนติกอัลกอริทึมสำหรับปัญหาที่มีหลายวัตถุประสงค์

3.1 เจนเนติกอัลกอริทึม

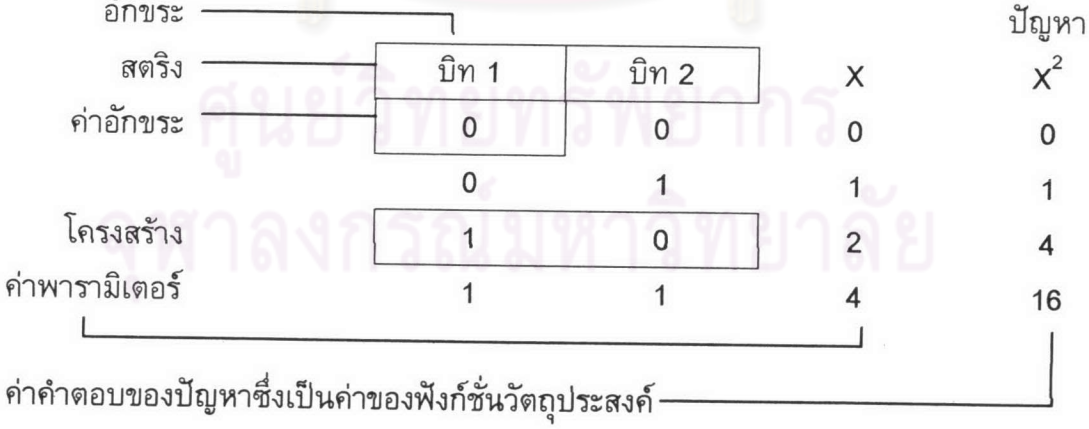
ในปัจจุบันการหาคำตอบของปัญหาบางประเภท เช่น ปัญหาการจัดสรรทรัพยากรที่มีอยู่อย่างจำกัดเพื่อให้เกิดประสิทธิภาพสูงสุด และปัญหาในการคำนวณต้นทุนต่ำสุด เป็นต้น สามารถหาคำตอบได้หลายวิธี วิธีการที่ง่ายที่สุดในการหาคำตอบคือวิธีการทางฮิวริสติกต่างๆ ซึ่งอาจได้คำตอบที่ไม่ดีนัก ปัจจุบันนักวิทยาศาสตร์ได้นำความรู้เกี่ยวกับทฤษฎีหรือกฎเกณฑ์ทางธรรมชาติ มาช่วยในการหาคำตอบหรือศึกษาวิจัย โดยมีเป้าหมายหลักในการใช้ประโยชน์ของความคงทน (Robustness) ต่อความไม่เที่ยงตรงแม่นยำ (Accuracy) ความไม่แน่นอน (Uncertainty) หรือความคลุมเครือของปัญหา (Vague) หลักการเหล่านี้สามารถพบได้จากวิธีการต่างๆ เช่น ระบบโครงข่ายประสาทเทียม (Neural Network) ฟัซซี่ลอจิก (Fuzzy Logic) (Zadeh, 1965) และ เจนเนติกอัลกอริทึม (Genetic Algorithms, GAs) (Goldberg, 1989) ปัญหาที่พบส่วนใหญ่เป็นปัญหาที่ไม่เที่ยงตรงและคลุมเครือ ซึ่งหากต้องการคำตอบที่เที่ยงตรงและมีความแน่นอนสูงมากก็ย่อมมีค่าใช้จ่ายที่สูงมาก ดังนั้นวิธีการที่สามารถแก้ปัญหาที่คลุมเครือโดยได้คำตอบที่ใกล้เคียงสามารถยอมรับได้ ใช้เวลาในการหาคำตอบไม่มากนัก และมีค่าใช้จ่ายพอประมาณ ย่อมดีกว่าวิธีที่ได้ความเที่ยงตรงสูงแต่มีค่าใช้จ่ายที่สูง วิธีการหาคำตอบที่ดีอย่างหนึ่งได้แก่วิธีการของเจนเนติกอัลกอริทึม โดยอาศัยทฤษฎีในการถ่ายทอดลักษณะต่างๆ ทางกรรมพันธุ์ไปสู่ลูกหลาน ซึ่งสามารถนำมาพัฒนาใช้ในการหาคำตอบที่ต้องการได้

3.1.1 พันธุศาสตร์กับเจนเนติกอัลกอริทึม

Mendel บิดาแห่งวิชาพันธุศาสตร์ ค้นพบว่าลักษณะต่าง ๆของสิ่งมีชีวิต 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 แสดงถึงการแก้ปัญหาในการหาค่าสูงสุดของฟังก์ชั่น $f(x)=x^2$ โดยที่ x มีค่าอยู่ระหว่าง [0,4] และค่าของ x ถูกแปลงให้อยู่ในรูปไบนารี่สตริง

***รูปที่ 3.1** การเปรียบเทียบลักษณะระหว่างเจนเนติกอัลกอริทึมกับลักษณะทางพันธุศาสตร์*

การแก้ปัญหาทางด้านคณิตศาสตร์ด้วยเจนเนติกอัลกอริทึม พารามิเตอร์ต่างๆจะถูกแปลงให้อยู่ในรูปของสตริง (String) หรือโครโมโซมประกอบด้วยอักขระ (Character) หรือ (Bit) แต่ละตำแหน่งของโครโมโซมจะเก็บค่าอักขระหรือค่าของบิท ที่แสดงโครงสร้างของแต่ละโครโมโซมที่ให้คำตอบของปัญหาแตกต่างกัน ดังรูปที่ 3.1ข ซึ่งเป็นการประยุกต์ใช้เจนเนติกอัลกอริทึม กับการแก้ปัญหาการหาค่าสูงสุดของฟังก์ชัน $f(x) = x^2$ โดยที่ x อยู่ในช่วง [0, 4] และสามารถสรุปความหมายทางพันธุศาสตร์เทียบกับเจนเนติกอัลกอริทึม ได้ดังตารางที่ 3.1

***ตารางที่ 3.1** เปรียบเทียบคำศัพท์ระหว่างพันธุศาสตร์และเจนเนติกอัลกอริทึม*

พันธุศาสตร์	เจนเนติกอัลกอริทึม (GAs)
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)

Darwin (1859) ได้เสนอความคิดการเกิดสปีชีส์ของสิ่งมีชีวิต (The Origin of Species) โดยเสนอหลักการของวิวัฒนาการที่ผ่านกระบวนการคัดเลือกตามธรรมชาติ แม้ในตอนแรกทฤษฎีจะเป็นที่โต้แย้งกันมากต่อมาก็ได้เป็นที่ยอมรับในหมู่นักวิทยาศาตร์ ซึ่งทฤษฎีดังกล่าวคือ

- สิ่งมีชีวิตแต่ละชนิดมีแนวโน้มที่จะถ่ายทอดลักษณะของมันไปสู่ลูกหลานของมัน
- ธรรมชาติทำให้สิ่งมีชีวิตมีลักษณะต่างๆกัน
- สิ่งมีชีวิตมีความเหมาะสม ซึ่งมีลักษณะที่เหมาะสมที่สุด มีแนวโน้มที่จะมีลูกหลานมากกว่าสิ่งมีชีวิตที่มีลักษณะไม่เหมาะสม ซึ่งจะทำให้ประชากรอยู่รอดต่อไป
- เมื่อระยะเวลาผ่านไปยาวนาน จะเกิดการกลายพันธุ์ (Variation) ขึ้น และเกิดสปีชีส์ใหม่ที่มีลักษณะเหมาะสมกับระบบนิเวศน์นั้น

3.1.2 ความหมายของเจนเนติกอัลกอริทึม

เจนเนติกอัลกอริทึมเป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานมาจากกระบวนการคัดเลือกทางธรรมชาติ (Natural Selection) และ กระบวนการคัดเลือกทางพันธุศาสตร์ (Natural Genetics Selection) โดยการคัดเลือกสตริง (String) ที่มีความเหมาะสมจากกลุ่มของสตริงทั้งหมดด้วยวิธีการสุ่ม จากการนำสตริงเหล่านี้ไปผ่านกระบวนการคัดเลือกสตริงที่มีความ

เหมาะสม ซึ่งสตริงที่มีความเหมาะสมนี้คือคำตอบที่ดีที่สุดหรือใกล้เคียงคำตอบที่ดีที่สุดเจนเนติกอัลกอริทึมไม่ใช่การสุ่มแบบง่ายๆ แต่มันเป็นการใช้ข้อมูลในอดีตอย่างมีประสิทธิภาพเพื่อพิจารณาจุดที่จะต้องค้นหาใหม่ โดยคาดหวังว่าสมรรถนะของการค้นหาจะดีขึ้น

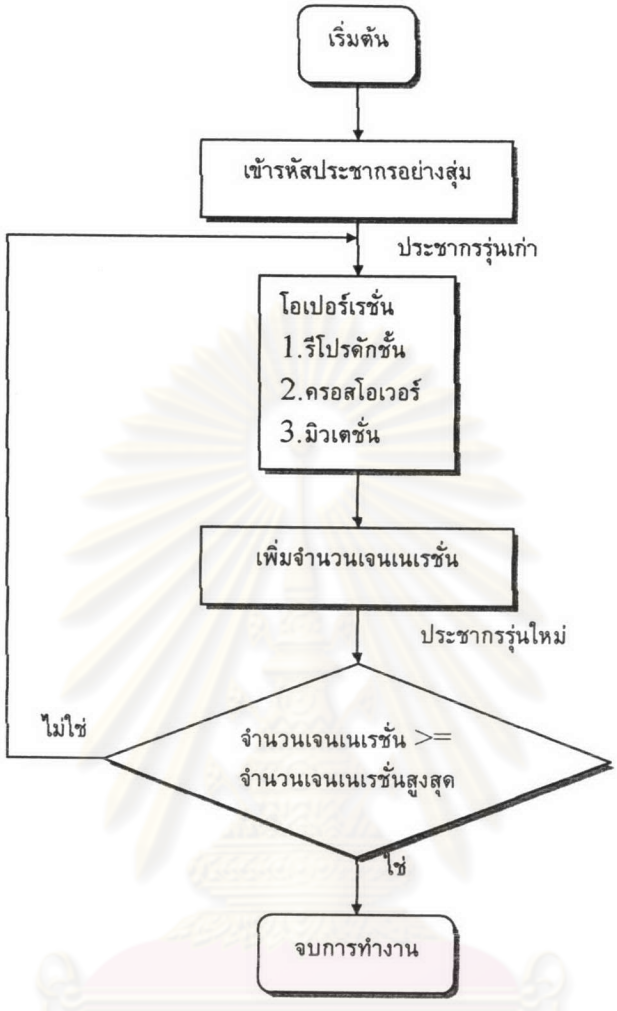
เจนเนติกอัลกอริทึมถูกพัฒนาขึ้นโดย Holland (1975) และคณะ โดยมีเป้าหมายในการวิจัย 2 อย่าง คือ ข้อแรก เพื่อสรุปและดัดแปลงการใช้กระบวนการทางธรรมชาติให้ถูกต้องมากที่สุด สองเพื่อออกแบบและสร้างซอฟแวร์ที่รักษากลไกที่สำคัญของธรรมชาติ และ GAs แตกต่างกับวิธีการค้นหาและการทำ Optimization แบบอื่นๆ คือ

- เจนเนติกอัลกอริทึม ทำงานโดยการเข้ารหัสสตริงเป็นชุดพารามิเตอร์
- เจนเนติกอัลกอริทึม เป็นการค้นหาจากทั้งประชากรไม่ใช่ค้นหาจากเพียงตำแหน่งๆ เดียว
- เจนเนติกอัลกอริทึม ใช้ฟังก์ชั่นเป้าหมายเป็นข้อมูลในการหาคำตอบ โดยไม่ใช้การหาอนุพันธ์หรือความรู้อื่น
- เจนเนติกอัลกอริทึม จะเป็นวิธี Probabilistic ไม่ใช่ Deterministic

3.2 เจนเนติกอัลกอริทึมอย่างง่าย (Simple Genetic Algorithms)

ขั้นตอนการทำเจนเนติกอัลกอริทึม อย่างง่าย (Simple Genetic Algorithms) หรือ SGA สามารถแสดงได้ ดังรูปที่ 3.2

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.2 ขั้นตอนของเจนเนติกอัลกอริทึมอย่างง่าย

3.2.1 การเข้ารหัส และการสร้างประชากรเริ่มต้นอย่างสุ่ม

ขั้นตอนแรกของเจนเนติกอัลกอริทึม คือ การเข้ารหัสหรือแปลงค่าพารามิเตอร์ให้อยู่ในรูปของสตริงที่มีความยาวแน่นอน ซึ่งวิธีการเข้ารหัสนี้ขึ้นอยู่กับรูปแบบของปัญหาแต่ละปัญหา สำหรับ SGA ใช้การเข้ารหัสแบบไบนารี่ (Binary Coding)

ตัวอย่างเช่น ต้องการหาค่าสูงสุดของฟังก์ชั่น $f(x) = x^2$ โดยที่ x มีค่าอยู่ระหว่าง [0,31] ในที่นี่ฟังก์ชั่นวัตถุประสงค์ (Objective Function) คือ f(x) หรือ x^2 ซึ่งวิธีการเข้ารหัสแบบไบนารี่ โดยแปลงค่าพารามิเตอร์ x ให้อยู่ในรูปไบนารี่ 5 บิทจะได้ 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 รีโปรดักชั่น (Reproduction) การครอสโอเวอร์ (Crossover) และ การมิวเตชั่น (Mutation) มีรายละเอียดดังต่อไปนี้

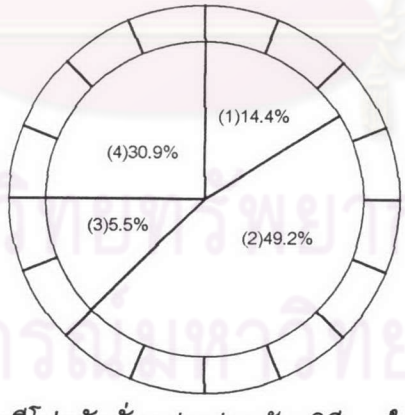
- **รีโปรดักชั่น (Reproduction)** คือ กระบวนการที่สตริงแต่ละตัวเลียนแบบค่าฟังก์ชันเป้าหมาย f(x) โดยที่ฟังก์ชั่นนี้อาจเป็นการวัด ผลตอบแทน ค่าอรรถประโยชน์ (Utility function) หรือสิ่งที่ต้องการให้เป็นค่าสูงสุด หรือค่าความเหมาะสม สตริงที่มีความเหมาะสมสูงกว่าก็จะมีความน่าจะเป็นในการสนับสนุนลูกหลานรุ่นต่อไปสูงด้วย ตัวปฏิบัติการนี้เกิดขึ้นจากกระบวนการคัดเลือกตามธรรมชาติตามทฤษฎีผู้อยู่รอดที่มีความเหมาะสม (Survival of Fittest) ของ ชาล์ส ดาร์วิน ประชากรที่มีความเหมาะสมในธรรมชาติจะมีความสามารถในการรอดพ้นจากผู้ล่า โรคภัยไข้เจ็บ อุปสรรคอื่นๆที่ต่อต้านการเจริญเติบโตเป็นผู้ใหญ่และสามารถสืบพันธุ์ต่อไปได้ ส่วนฟังก์ชันเป้าหมายจะเป็นสิ่งที่ใช้พิจารณาว่าสตริงที่สร้างขึ้นจะมีชีวิตอยู่หรือตายจากไป

ตัวปฏิบัติการรีโปรดักชั่นสามารถสร้างขึ้นได้หลายวิธี วิธีที่ง่ายวิธีหนึ่งคือสร้างจากวงล้อรูลเล็ทที่มีจำนวนช่องเท่ากับจำนวนประชากรสตริง และขนาดของช่องก็เป็นสัดส่วนกับค่าความเหมาะสม ดังรูปที่ 3.3 และค่าความเหมาะสมของฟังก์ชันเป้าหมายของประชากรทั้งสี่แสดงอยู่ในตาราง 3.2

ค่าความเหมาะสมทั้งหมดโดยรวมจะได้ 1170 และค่ารายละเอียดต่างๆแสดงดังในตารางที่ 3.2 แสดงถึงวงล้อรูลเล็ทสำหรับการรีโปรดักชั่น ซึ่งสร้างจากสัดส่วนของค่าฟิตเนสของสตริงทั้งหมด เช่นสตริงหมายเลข 1 มีค่าความเหมาะสมเป็น 169 หรือ 14.4% (169/1170) ของค่าฟิตเนสโดยรวมของทั้งประชากร ในการทำการรีโปรดักชั่นจะหมุนวงล้อเป็นจำนวน 4 ครั้ง หรือเท่ากับจำนวนสตริง เช่น สตริงหมายเลข 1 มีค่าเป็น 169 คิดเป็น 14.4% ของค่าความเหมาะสมทั้งหมด ดังนั้นเมื่อหมุนรูลเล็ท 1 ครั้งก็จะมีความน่าจะเป็นที่ถูกเลือกเท่ากับ 0.144 ในการหมุนรูลเล็ทแต่ละครั้งจะได้ตัวแทนในการสืบพันธุ์ (Reproduction Candidate) สตริงที่มีความเหมาะสมสูงจะถูกคัดเลือกสำหรับการสืบพันธุ์การรีโปรดักชั่นสำหรับสตริงลูกหลานในรุ่นต่อไป เมื่อสตริงมีรูปร่างที่แน่นอนแล้วก็จะถูกส่งไปเข้าเมทติ้งพูล (Matting Pool) เพื่อที่จะผ่านกระบวนการของตัวปฏิบัติการอื่นต่อไป

ตารางที่ 3.2 *กลุ่มประชากรตัวอย่างและค่าความเหมาะสม*

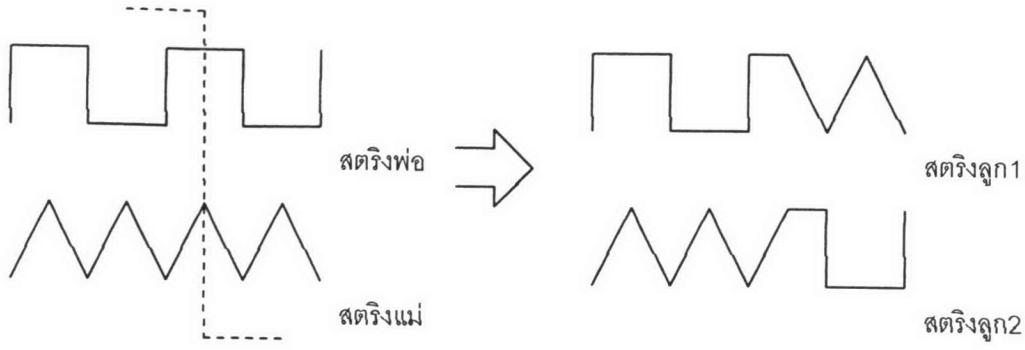
No.	สตริง	ค่าความเหมาะสม	% โดยรวม
1	01101	169	14.40
2	11000	576	49.20
3	01000	64	5.50
4	10011	361	30.90
รวม		1170	100.00



รูปที่ 3.3 *การรีโปรดักชั่นอย่างง่ายด้วยวิธีการใช้วงล้อรูลเล็ท*

- **ครอสโอเวอร์ (Crossover)** หลังจากประชากรทั้งหมดผ่านกระบวนการรีโปรดักชั่นแล้ว จะทำการจับคู่สมาชิกในเมทติ้งพูลหรือกลุ่มประชากรทั้งหมดอย่างสุ่มและทำการไขว้สลับค่าที่อยู่หลังตำแหน่งที่เลือกไว้จากการสุ่มหรือ ทำการแลกเปลี่ยนสวนกัน ตัวอย่างการครอสโอเวอร์แสดงดังรูปที่ 3.4

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



***รูปที่ 3.4** การครอสโอเวอร์อย่างง่ายเพื่อให้เห็นถึงการเปลี่ยนแปลงสตริง และการแลกเปลี่ยนข่าวสารโดยเลือกตำแหน่งไขว้แบบสุ่ม*

การเลือกตำแหน่งที่จะทำการครอสโอเวอร์ จะทำโดยการสุ่มค่าที่เป็นจำนวนเต็มตำแหน่งที่ **k** ช่วงของสตริงที่เลือกจะอยู่ในช่วง **[1, t-1]** โดยที่ **t** คือตำแหน่งสุดท้ายของสตริง และสตริงใหม่ทั้งสองก็จะมีการสลับอักขระตั้งแต่ตำแหน่งที่ **k+1** จนถึง **t** ยกตัวอย่างเช่น พิจารณาสตริง A_1, A_2 จากประชากรเริ่มต้น

$A_1 = 0110 \mid 1$
$A_2 = 1100 \mid 0$

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

$A'_1 = 01100$
$A'_2 = 11001$

- **มิวเตชั่น (Mutation)** มิวเตชั่นเป็นสิ่งที่จำเป็นถึงแม้ว่ารีโปรดักชั่นและครอสโอเวอร์ช่วยให้การค้นหาเป็นไปอย่างมีประสิทธิภาพ ในบางครั้งก็มีการสูญเสียส่วนที่สำคัญไป (ค่า **1** หรือ **0** ในบางตำแหน่ง) การมิวเตชั่นจะป้องกันส่วนที่สูญเสียที่ไม่อาจเรียกคืนได้ **(Irrecovery Loss)** ในบางครั้งการหาคำตอบของ SGA คำตอบอาจติดอยู่ใน **Local Optima** การมิวเตชั่นด้วยอัตราส่วนที่เหมาะสมจะทำให้คำตอบสามารถหลุดออกจาก **Local Optima** หรืออาจกล่าวได้ว่าโอเปอร์เรเตอร์ของการมีมิวเตชั่นเป็นการเปลี่ยนแปลงค่าตำแหน่งสตริงแบบสุ่ม จากปัญหาที่พิจารณาค่าจะเปลี่ยนแปลงจาก **0** เป็น **1** หรือ **1** เป็น **0** โดยการเลือกตำแหน่งที่จะทำการมิวเตชั่นอย่างสุ่ม อัตราการมิวเตชั่นในธรรมชาติจะมีค่าค่อนข้างต่ำ ในการนำไปใช้งานจะต้องมีการพิจารณาอย่างเหมาะสม

3.2.4 ประชากรรุ่นใหม่ (New population)

สตริงทั้งหมดที่ได้จากกระบวนการของ GAs เรียกว่า ประชากรรุ่นใหม่ หรือเจนเนอเรชั่น (Generation) รุ่นใหม่ ซึ่งจะกลายเป็นประชากรรุ่นเก่า สำหรับการดำเนินการต่อไป กระบวนการของ SGA จะทำซ้ำไปเรื่อยๆจนกว่าจำนวนเจนเนอเรชั่นจะมากกว่าจำนวนเจนเนอเรชั่นที่กำหนดไว้สูงสุด

Surech (1995) ได้พิจารณาถึงการหาขนาดของประชากร จากอัตราส่วนของวิธีการของคำตอบทั้งหมดที่เป็นไปไม่ได้ และอัตราส่วนของวิธีการที่ทั้งหมดของคำตอบที่เป็นไปได้

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

จากสมการที่ (3.1) ให้

n คือ ขนาดของปัญหา

n! คือ จำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงที่เป็นไปได้

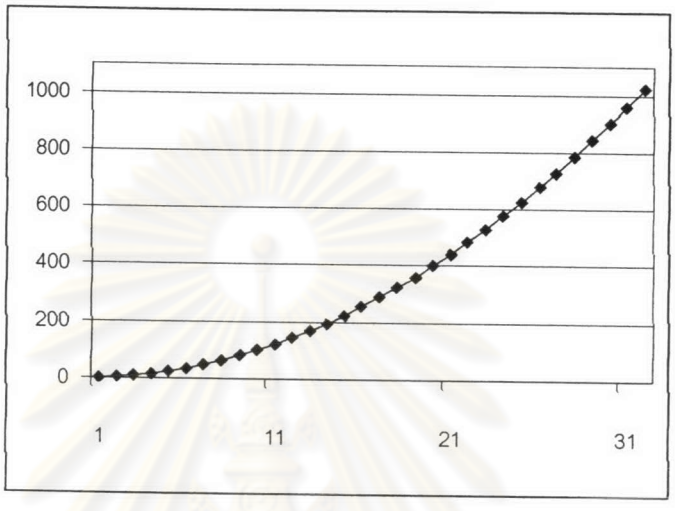
n^n คือ จำนวนวิธีการที่จัดเรียงหรือจำนวนวิธีการจัดเรียงที่เป็นไปไม่ได้ (*n.n…n*)

สามารถสรุปได้ว่า ความน่าจะเป็นในการสร้างประชากรคำตอบอย่างสุ่ม มีค่าเป็นศูนย์เมื่อ *n* มีค่ามากขึ้น สมมติว่า *n* มีค่าเป็น 8 ดังนั้น 8! / 8^8 = 1/ 416.1 = $2.403*10^{-3}$ หรืออาจกล่าวได้ว่า โอกาสที่จะได้คำตอบที่ถูกต้องเป็น 1 ใน 416 ของคำตอบที่เป็นไปไม่ได้ ถ้ากำหนดจำนวนประชากรเป็น 100 และทำการคำนวณเพียงแค่ 1 เจนเนอเรชั่นก็ไม่อาจคาดได้ว่าจะได้คำตอบที่ดี การกำหนดจำนวนประชากรเริ่มต้นและจำนวนเจนเนเรชั่นทั้งหมดจึงมีผลในการหาคำตอบ

3.3 ตัวอย่างการใช้เจนเนติกอัลกอริทึมในการหาคำตอบของฟังก์ชั่น

เนื้อหาในส่วนนี้ จะเป็นการประยุกต์ใช้เจนเนติกอัลกอริทึมในการแก้ปัญหา Optimization หาค่าสูงสุดของฟังก์ชัน f(x) = x^2 ทีละขั้นตอน โดย x เป็นตัวแปรที่มีค่าเปลี่ยนแปลงระหว่าง 1 ถึง 31 ดังรูปที่ 3.5 แสดงถึงลักษณะฟังก์ชัน f(x) สำหรับปัญหานี้ตัวแปร x จะถูกเข้ารหัสให้เป็นไบนารี่ ที่มีความยาวสตริง 5 บิท

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 ฟังก์ชั่นวัตถุประสงค์ $f(x) = x^2$

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

3.3 ก) การสุ่มสตริงเริ่มต้นและการรีโปรดักชั่น

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

3.3 ข) การครอสโอเวอร์

เมทติ้งพูล หลังจากการ รีโปรดักชั่น	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 บิท)
- รีโปรดักชั่นได้จากการหมุนวงล้อรูลเล็ท
- ครอสโอเวอร์ได้จากการโยนเหรียญสองเหรียญแล้วทำการถอดรหัส (TT = 00_2 = 0 = ตำแหน่งที่ไขว้คือ 1, HH = 11_2 = 3 = ตำแหน่งที่ไขว้คือ 4)
- ความน่าจะเป็นของครอสโอเวอร์กำหนดให้เป็นหนึ่ง p_c = 1.0
- ความน่าจะเป็นของมิวเตชั่นเป็น 0.001, p = 0.001, expected mutation = 5*4*0.001 =0.2 ไม่มีค่า expected mutation ระหว่างประชากรเดี่ยว

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

ตัวปฏิบัติการสุดท้ายคือมิวเตชั่นซึ่งจะเปลี่ยนค่าเป็นบิทต่อบิท สมมุติความน่าจะเป็นของการมิวเตชั่นในการทดสอบเป็น 0.001 ตำแหน่งที่จะเปลี่ยนแปลงทั้งหมดมี 20 บิท (ได้จากจำนวนสตริง*จำนวนบิทของสตริงแต่ละตัว 5*4=20) เพราะฉะนั้นตำแหน่งบิทที่จะมิวเตชั่นของประชากรรุ่นนี้คือ 20 * .001 = 0.02 บิท จากการคำนวณจะเห็นได้ว่าไม่มีบิทใดต้องทำการมิวเตชั่นสำหรับค่าความน่าจะเป็นนี้ นั่นก็คือไม่มีบิทใดที่จะต้องเปลี่ยนค่าจาก 1 เป็น 0 หรือ 0 เป็น 1 สำหรับประชากรรุ่นนี้ แต่สมมติว่าถ้าตำแหน่งบิทที่จะมิวเตชั่นของประชากรรุ่นนี้คือ 5 ดังนั้นตำแหน่งบิทที่ 5 จะต้องทำการเปลี่ยนค่าจาก 0 เป็น 1 หรือ 1 เป็น 0

หลังจากผ่านการรีโปรดักชั่น ครอสโอเวอร์และมิวเตชั่น ประชากรรุ่นใหม่ก็พร้อมที่จะถูกทดสอบ โดยทำการเข้ารหัสสตริงใหม่คำนวณหาค่า x และค่าฟังก์ชัน f(x) ตารางที่ 3.3 ข. แสดงถึงผลจากการทดลอง จะเห็นได้ว่ากระบวนการที่เกี่ยวข้องกับความน่าจะเป็นทำให้ค่าสมรรถนะดีขึ้น 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)

- การเข้ารหัสของปัญหาผิดพลาด ทำให้ GAs หาคำตอบผิดพลาด
- ขีดจำกัดของจำนวนประชากร ในทางทฤษฎีแล้วมีค่าเป็นอนันต์
- ขีดจำกัดของจำนวนเจนเนอเรชั่น ในทางทฤษฎีแล้วมีค่าเป็นอนันต์

ค่าพารามิเตอร์เหล่านี้ไม่สามารถกำหนดให้เป็นอนันต์ได้ในทางปฏิบัติ เนื่องจากข้อจำกัดต่างๆของคอมพิวเตอร์

3.4 เจนเนติกอัลกอริทึมแบบหลายวัตถุประสงค์

ในอดีตได้มีการคิดค้นวิธีการ เพื่อใช้หาคำตอบของปัญหาการหาค่าที่ดีที่สุดทางวิศวกรรมเป็นจำนวนมาก แต่ปัญหาที่ศึกษาส่วนใหญ่มักเป็นปัญหาในอุดมคติ วิธีการส่วนมาก

มักพิจารณาวัตถุประสงค์ในการแก้ปัญหาเพียงอย่างเดียว แต่อย่างไรก็ตามในความเป็นจริงแล้ว ปัญหาที่พบส่วนใหญ่มักเป็นปัญหาที่มีหลายวัตถุประสงค์ ดังนั้นจึงต้องอาศัยประสบการณ์ของผู้ออกแบบเป็นหลักในการตัดสินเลือกคำตอบที่ดีที่สุด

ในช่วงหลายปีที่ผ่านมา ได้มีการคิดค้นพัฒนาเทคนิคทางคณิตศาสตร์ใหม่ขึ้นมาเพื่อใช้กับปัญหาที่มีหลายวัตถุประสงค์ ส่วนมากมักเป็นการหาคำตอบที่ดีที่สุดออกมาเพียงคำตอบเดียวมากกว่าการหาคำตอบที่เป็นไปได้หลายๆคำตอบ แล้วจึงให้ผู้ออกแบบตัดสินใจเลือกภายหลัง วิธีการหนึ่งที่ได้รับความสนใจอย่างมากคือวิธีการของเจนเนติกอัลกอริทึม ซึ่งงานวิจัยหลายชิ้นได้เสนอแนวทางการพัฒนาวิธีการของเจนเนติกอัลกอริทึมอย่างง่ายเพื่อให้สามารถนำไปใช้จัดการกับปัญหาแบบหลายวัตถุประสงค์ได้

วิธีการเจนเนติกอัลกอริทึมสำหรับปัญหาแบบหลายวัตถุประสงค์ที่มีการพัฒนาขึ้นมานั้นมีทั้งแบบที่ให้คำตอบที่ดีที่สุดค่าเดียว และแบบที่ให้หลายคำตอบที่เป็นไปได้ ทั้งนี้ขึ้นกับความเหมาะสมและความต้องการของผู้ออกแบบ แต่อย่างไรก็ตาม ถึงแม้จะมีการพัฒนาเจนเนติกอัลกอริทึมเพื่อจัดการกับปัญหาหลายวัตถุประสงค์มากมายหลายวิธี แต่วิธีการที่พัฒนาขึ้นนี้ไม่ได้ถูกกำหนดว่าวิธีใดเป็นวิธีที่เหมาะสมที่สุด เนื่องจากเป็นยังเป็นเพียงการพัฒนาในขั้นต้นเท่านั้น

การพัฒนาส่วนมาก เป็นการพัฒนาในส่วนของการคัดเลือกคำตอบที่มีความเหมาะสมที่สุด ในขั้นตอนของการประเมินค่า (Evaluate) และคัดเลือกคำตอบ (Selection) วิธีการที่มีสามารถแบ่งออกเป็น 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

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

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 Niched 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)

วิธีการให้น้ำหนัก (Weight Sum Approach) วิธีการนี้จะทำการรวมวัตถุประสงค์ต่างๆ เข้าเป็นวัตถุประสงค์เดียว โดยทำการเพิ่มสัมประสิทธิ์น้ำหนักของแต่ละวัตถุประสงค์ สามารถแสดงดังสมการ(3.2)

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

$f_i(x)$ คือ ค่าฟังก์ชั่นวัตถุประสงค์ต่างๆ
W_i คือค่าสัมประสิทธิ์ที่ให้แต่ละวัตถุประสงค์ ซึ่ง

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

วิธีการนี้มีข้อดีคือใช้งานง่าย ไม่ซับซ้อน แต่ก็มีข้อเสียคือ กำหนดน้ำหนักในแต่ละวัตถุประสงค์ให้เหมาะสมได้ยาก ดังนั้นเพื่อให้การกำหนดน้ำหนักมีความถูกต้อง ผู้กำหนดนั้นควรจะมีความรู้ ความเข้าใจในปัญหา และทราบถึงความจำเป็นในแต่ละวัตถุประสงค์เป็นอย่างดี

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

I21229958

3.5 สรุปท้ายบท

เจนเนติกอัลกอริทึมเป็นวิธีการค้นหาคำตอบวิธีหนึ่ง โดยมีพื้นฐานจากกระบวนการคัดเลือกทางธรรมชาติ ข้อดีของเจนเนติกอัลกอริทึมเมื่อเปรียบเทียบกับการค้นหาแบบอื่นคือมีความคงทนต่อความไม่เที่ยงตรงแม่นยำและความไม่แน่นอนหรือคลุมเครือของปัญหา และสามารถควบคุมได้ โดยมีความน่าเชื่อถือสูงและค่าใช้จ่ายต่ำ

วิธีการค้นหาของเจนเนติกอัลกอริทึม จะแตกต่างกับวิธีการค้นหาและการทำ Optimization แบบอื่นๆ คือ

- เจนเนติกอัลกอริทึม จะใช้งานโดยการเข้ารหัสสตริงเป็นชุดพารามิเตอร์
- เจนเนติกอัลกอริทึม เป็นการค้นหาจากทั้งประชากรไม่ใช่เพียงตำแหน่งๆเดียว
- เจนเนติกอัลกอริทึม ใช้ฟังก์ชั่นเป้าหมายเป็นข้อมูลในการหาคำตอบ โดยไม่ใช้การหาอนุพันธ์หรือความรู้อื่นๆ
- เจนเนติกอัลกอริทึม จะเป็นวิธี Probabilistic ไม่ใช่ Deterministic

โอเปอร์เรเตอร์ต่างๆของ GAs ได้แก่

- รีโปรดักชั่น คือกระบวนการคัดเลือกสตริงมีความเหมาะสมสูงเพื่อเป็นคำตอบเริ่มต้นให้กับประชากรรุ่นต่อไป โดยอาศัยทฤษฎีของ ชาร์ล ดาร์วินที่ว่าสิ่งมีชีวิตที่แข็งแรงกว่ามีโอกาสอยู่รอดในสภาวะนั้นๆได้มากกว่า
- การครอสโอเวอร์ คือ กระบวนการสร้างสตริงลูกหลานใหม่ จากสตริงพ่อแม่
- การมิวเตชั่นคือ กระบวนการที่ช่วยปรับปรุงสตริงให้ดีขึ้นหรืออาจจะเลวลง โดยการเปลี่ยนแปลงค่าในบางตำแหน่งของสตริง เพื่อให้เกิดสตริงใหม่

พารามิเตอร์ต่างๆของ GAs ได้แก่

- จำนวนประชากร (Population size)
- จำนวนเจนเนอเรชั่น (Number of generation)
- วิธีการรีโปรดักชั่น (Reproduction type)
- วิธีการครอสโอเวอร์ (Crossover type) และค่าความน่าจะเป็นของการครอสโอเวอร์ (Probability of Crossover,Pc)
- วิธีการการมิวเตชั่น (Mutation type) และค่าความน่าจะเป็นของการมิวเตชั่น (Probability of Mutation,Pm)

สามารถประยุกต์เจนเนติกอัลกอริทึมเพื่อแก้ปัญหาแบบหลายวัตถุประสงค์ได้ โดยวิธีการที่นิยมใช้และได้ผลดีวิธีหนึ่งคือการให้น้ำหนักในแต่ละปัจจัย (Weight Sum Approach)