บทที่ 2

กำหนดการเชิงพันธุกรรม

ในบทนี้กล่าวถึงรายละเอียดของกำหนดการเชิงพันธุกรรม โดยกล่าวถึงหลักการทั่วๆ ไปของกำหนดการเชิงพันธุกรรม ขั้นตอนการค้นหาคำตอบในวิธีกำหนดการเชิงพันธุกรรม การทำงานของตัวปฏิบัติการทางพันธุกรรมและสรุปท้ายบท

2.1 หลักการทั่วไป

กำหนดการเชิงพันธุกรรม (Genetic Programming) เป็นวิธีการเรียนรู้ และค้นหาคำตอบของปัญหา ได้พัฒนามาจากขั้นตอนวิธีเชิงพันธุกรรม (Genetic Algorithm) หรือ GA ซึ่งคิดค้นโดย Holland (1975) ข้อแตกต่างระหว่างขั้นตอนวิธีเชิงพันธุกรรม และกำหนดการเชิงพันธุกรรม คือลักษณะการแทนคำตอบ โดยขั้นตอนวิธีเชิงพันธุกรรมจะแทนลักษณะคำตอบในรูปสายอักขระที่มีขนาดคงที่ แต่กำหนดการเชิงพันธุกรรมจะแทนลักษณะคำตอบในรูปโปรแกรมคอมพิวเตอร์ ซึ่งสามารถแทนด้วยโครงสร้างแบบต้นไม้

กำหนดการเชิงพันธุกรรม ถูกคิดค้นโดย Koza (1992) เป็นวิธีการเรียนรู้เพื่อแก้ปัญหาโดยเลียนแบบการวิวัฒนาการของสิ่งมีชีวิต วิวัฒนาการในธรรมชาติของสิ่งมีชีวิตไม่มีแบบแผนและเป้าหมายที่แน่นอน แต่กระบวนการทางธรรมชาติจะผลักดันให้สิ่งมีชีวิตแต่ละตัวแข่งขันกันเพื่อดำรงชีวิตอยู่ในสภาพแวดล้อมของตนเอง สิ่งมีชีวิตใดที่อ่อนแอก็จะอยู่ไม่ได้แล้วตายไป ส่วนสิ่งมีชีวิตที่แข็งแรงกว่าสามารถที่จะมีชีวิตอยู่รอดได้ก็จะคงอยู่และมีวิวัฒนาการของตนเองต่อไป แต่สำหรับการวิวัฒนาการคำตอบซึ่งอยู่ในรูปโปรแกรมคอมพิวเตอร์ในวิธีกำหนดการเชิงพันธุกรรม มีเป้าหมายเพื่อให้ได้โปรแกรมคำตอบที่มีความสามารถในการแก้ปัญหาที่กำหนดได้ โดยโปรแกรมที่มีความเหมาะในการแก้ปัญหาสูงกว่าจะได้รับการวิวัฒนาการไปจนได้เป็นโปรแกรมที่เป็นคำตอบของปัญหา เราจะเรียกโปรแกรมคอมพิวเตอร์ที่อยู่ในระหว่างขั้นตอนการวิวัฒนาการว่า **ผลเฉลย** (Individual) ส่วนผลเฉลยที่สามารถทำงานแก้ปัญหาที่กำหนดได้จะเรียกว่าเป็น **คำตอบ** (Solution) ของปัญหา

การค้นหาคำตอบโดยกำหนดการเชิงพันธุกรรม แบ่งเป็นขั้นตอนหลักคือ การสร้างประชากรผลเฉลยเริ่มต้น การประเมินค่าความเหมาะของผลเฉลย การสร้างประชากรของผลเฉลยรุ่นใหม่ และการหาคำตอบ ซึ่งใน

ส่วนของการประเมินค่าความเหมาะ และการสร้างประชากรของผลเฉลยรุ่นใหม่ จะเรียกว่า เป็น **กระบวนการวิวัฒนาการ** ซึ่งสามารถอธิบายได้ในรูปของรหัสเทียม (Pseudo Code) ดังรูปที่ 2.1

```
จำนวนรุ่น = 0 ;
สร้างประชากรผลเฉลยเริ่มต้นตามจำนวนประชากรที่กำหนด ;
While ( จำนวนรุ่น <= จำนวนรุ่นที่กำหนด หรือ ยังไม่พบคำตอบที่ต้องการ )
        ประเมินค่าความเหมาะของผลเฉลยแต่ละตัวในการแก้ปัญหาที่กำหนด ;
        เลือกผลเฉลยตามลำดับค่าความเหมาะมา n ตัว ;
        ใช้ตัวดำเนินการลีบพันธุ์กับผลเฉลย n ตัวเหล่านั้น นำไปเป็นประชากรรุ่นใหม่ ;
        จำนวนผลเฉลย = n ;
        While ( จำนวนผลเฉลย != จำนวนประชากรที่กำหนด )
                สร้างประชากรผลเฉลยในรุ่นใหม่ขึ้นมาจากผลเฉลย n ตัวเหล่านั้น
                โดยเลือกตัวดำเนินการดังต่อไปนี้ ;
                Case ( การไขว้เปลี่ยน )
                        เลือกผลเฉลย 2 ตัวอย่างสุ่มจากผลเฉลย n ตัวเหล่านั้น ;
                        สุ่มตำแหน่งที่จะทำการไขว้เปลี่ยน ;
                        ทำการแลกเปลี่ยนบางส่วนของผลเฉลย ได้ผลเฉลยขึ้นมา 2 ตัว ;
                        จำนวนผลเฉลย += 2 ;
                Case ( การกลาย )
                        เลือกผลเฉลยมา 1 ตัวอย่างสุ่มจากผลเฉลย n ตัวเหล่านั้น ;
                        สุ่มตำแหน่งที่จะทำการกลาย ;
                        ทำการแทนที่ส่วนนั้น ด้วยโปรแกรมเล็กๆ ได้ผลเฉลยขึ้นมาอีก 1 ตัว ;
                        จำนวนผลเฉลย += 1 ;
        End While
        จำนวนรุ่น++ ;
End While
ให้โปรแกรมที่มีค่าความเหมาะดีที่สุดเป็นคำตอบ ;
```

รูปที่ 2.1 Pseudo Code แสดงกระบวนการวิวัฒนาการในการหาคำตอบของวิธีกำหนดการเชิงพันธุกรรม

2.2 การสร้างประชากรผลเฉลยเริ่มต้น

ในขั้นตอนนี้จะทำการสร้างประชากรของผลเฉลยเริ่มต้นโดยใช้วิธีการสุ่ม โดยผลเฉลยแต่ละตัวอยู่ในรูปโปรแกรมคอมพิวเตอร์ โดยโปรแกรมแต่ละตัวเหล่านี้อาจจะทำงานแก้ปัญหาที่กำหนดนั้นได้หรือไม่ก็ได้ ลักษณะโครงสร้างของโปรแกรมจะอยู่ในรูปแบบของต้นไม้ที่มีขนาดและความสูงไม่จำกัด ประกอบด้วยส่วนประกอบ 2 ชนิดคือ ฟังก์ชัน (Function) และเทอร์มินอล (Terminal)

ฟังก์ชัน อาจเป็นฟังก์ชันทางคณิตศาสตร์ ฟังก์ชันทางตรรก หรือเป็นฟังก์ชันเฉพาะสำหรับปัญหานั้นๆ เทอร์มินอล อาจเป็นค่าคงที่ ตัวแปรหรือคำสั่งที่ส่งผลโดยตรงกับปัญหานั้น ข้อแตกต่างระหว่างฟังก์ชันและเทอร์มินอลคือ ฟังก์ชันจะต้องการตัวแปรหรืออาร์กิวเมนต์เพื่อช่วยในการทำงาน ในขณะที่เทอร์มินอลจะไม่ต้องการอาร์กิวเมนต์ อาจกล่าวได้ว่าเทอร์มินอลคือฟังก์ชันที่ไม่ต้องการอาร์กิวเมนต์ มีข้อสังเกตคือ อาร์กิวเมนต์ของฟังก์ชันสามารถเป็นฟังก์ชันอีกฟังก์ชันหนึ่งก็ได้ ดังนั้นฟังก์ชันจึงสามารถรับค่าส่งคืน (Return Value) ของอาร์กิวเมนต์ที่เป็นเทอร์มินอลหรืออาร์กิวเมนต์ที่เป็นฟังก์ชันก็ได้ จากลักษณะดังกล่าวการกำหนดฟังก์ชันและเทอร์มินอลจึงต้องมีรูปแบบที่เป็นมาตรฐานไปในทางเดียวกันเพื่อให้ฟังก์ชันทุกฟังก์ชันสามารถใช้งานร่วมกับเทอร์มินอลทุกเทอร์มินอลได้ (Closure Property) ดังที่กล่าวใน Koza (1992)

ตัวอย่างของฟังก์ชัน ได้แก่

- ตัวดำเนินการคณิตศาสตร์ เช่น + , - , * , /
- ตัวดำเนินการเงื่อนไข เช่น IF<= , IF< , IF=, IF-AND, IF-OR
- ตัวดำเนินการตรรกศาสตร์ เช่น AND , OR , NOT
- ตัวแสดงการวนซ้ำ เช่น DO-WHILE
- ฟังก์ชันเฉพาะสำหรับปัญหานั้นๆ เช่น IF-FOOD-AHEAD, IF-SEE-OBSTACLE เป็นต้น

และตัวอย่างของเทอร์มินอล เช่น

- ค่าคงที่ เช่น 0.5, 1, 5
- ตัวแปร เช่น x, a, b, sensor-reading
- คำสั่งที่มีผลเฉพาะสำหรับปัญหานั้นๆ เช่น TurnLeft, Forward, SenseRight เป็นต้น

พิจารณาเซตของฟังก์ชัน

F = { + , - , * , / }

และเซตของเทอร์มินอล

T = { x , y , z }

เมื่อ x, y และ z เป็นตัวแปรแบบตัวเลข ทำหน้าที่เป็นอาร์กิวเมนต์ให้กับฟังก์ชัน

อย่างไรก็ตามเราอาจจะทำการรวมทั้งเซตของฟังก์ชัน และเซตของเทอร์มินอลเข้าด้วยกันเป็นเซต C ดังนี้

$C = F \cup T = \{ +, -, *, /, x, y, z \}$

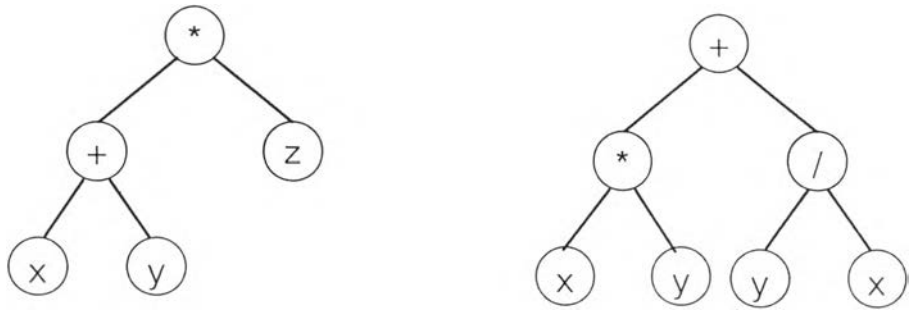
โดยที่เราสามารถแยกสมาชิกในเซต C ว่าเป็นเทอร์มินอลหรือฟังก์ชันได้ โดยพิจารณาจากจำนวนของอาร์กิวเมนต์ที่เป็นศูนย์ ซึ่งสมาชิกแต่ละตัวในเซต C มีจำนวนอาร์กิวเมนต์เป็น 2,2,2,2,0,0 และ 0 ตามลำดับ

สำหรับการเขียนสัญลักษณ์แทนผลเฉลยซึ่งเป็นโปรแกรมคอมพิวเตอร์ที่อยู่ในรูปต้นไม้ มักนิยมเขียนอยู่ในรูปแบบที่มีลักษณะเป็นนิพจน์เชิงสัญลักษณ์ (Symbolic Expression) ตัวอย่างของผลเฉลยหรือโปรแกรมคอมพิวเตอร์ เช่น

(* (+ x y) z)

(+ (* x y) (/ y x))

สามารถแทนโปรแกรมคอมพิวเตอร์ทั้งสองข้างต้น เป็นโครงสร้างต้นไม้ ดังรูปที่ 2.2



*รูปที่ 2.2 ต้นไม้ที่แทนโปรแกรมคอมพิวเตอร์ (* (+ x y) z)ซ้าย และ (+ (* x y) (/ y x))ขวา*

ผลจากขั้นตอนนี้จะได้ผลเฉลยในรูปโปรแกรมคอมพิวเตอร์เป็นจำนวนมากตามจำนวนที่ได้กำหนดไว้ ซึ่งจะมีลักษณะ และจำนวนของฟังก์ชัน และ เทอร์มินอลที่แตกต่างกันไปเนื่องมาจากการสุ่ม โดยโปรแกรมแต่ละตัวเหล่านี้อาจจะทำงานแล้วได้ผลลัพธ์ใกล้เคียงหรือไม่ใกล้เคียงกับคำตอบของปัญหานั้นก็ได้

2.3 <u>การประเมินค่าความเหมาะของผลเฉลย</u> (Fitness Evaluation)

หลังจากที่ได้ประชากรของผลเฉลยขึ้นมาเป็นจำนวนมากแล้ว ขั้นตอนต่อไปเพื่อหาว่าผลเฉลยใดมีประสิทธิภาพในการแก้ปัญหาที่กำหนด เพื่อนำไปสร้างประชากรของผลเฉลยในรุ่นใหม่ จะทำการคำนวณค่าความเหมาะ (Fitness Value) โดยใช้ฟังก์ชันวัดความเหมาะ (Fitness Function) ที่กำหนดขึ้นมาให้เหมาะสมกับปัญหานั้นๆ การนิยามฟังก์ชันวัดความเหมาะให้ดีเป็นส่วนที่ยาก และสำคัญในวิธีกำหนดการเชิงพันธุกรรม

ค่าความเหมาะนี้จะใช้เป็นเกณฑ์ในการตัดสินว่าประชากรตัวใดมีความสามารถในการแก้ปัญหาและสมควรที่จะถูกคัดเลือกให้อยู่รอด และขยายพันธุ์ต่อไป ดังนั้นเพื่อให้ประชากรที่ดีมีความสามารถในการแก้ปัญหาได้อยู่รอดต่อไปมากๆ จึงจำเป็นต้องมีการกำหนดฟังก์ชันวัดความเหมาะให้เหมาะสมกับปัญหานั้นมากที่สุด เพราะค่าความเหมาะนี้เป็นข้อมูลเพียงตัวเดียวป้อนกับให้กับระบบวิธีกำหนดการพันธุกรรม ซึ่งเป็นส่วนสำคัญในการวิวัฒนาการของคำตอบ

ค่าความเหมาะ เป็นค่าที่แสดงถึงความถูกต้องหรือประสิทธิภาพของโปรแกรม หรือผลเฉลย การวัดค่านี้กระทำโดยการนำโปรแกรมแต่ละตัวเหล่านั้นไปทดสอบแก้ปัญหาที่กำหนด ซึ่งอาจใช้สภาพจำลองในเครื่องคอมพิวเตอร์ (Simulation) หรือเป็นสภาพในโลกจริง การประมวลผลโปรแกรมในต้นไม้นั้น ทำได้ 2 ลักษณะ ดังที่กล่าวใน Raik และ Browne (1996) คือ

1) ทำการประมวลผลโปรแกรมจากตำแหน่งรากของต้นไม้ แล้วใช้ค่าส่งคืนจากรากของต้นไม้เป็นค่าส่งคืนของโปรแกรม

2) ทำการประมวลผลโปรแกรมจนกระทั่งไปถึงตำแหน่งของเทอร์มินอล ซึ่งเทอร์มินอลอาจเป็นการกระทำอย่างใดอย่างหนึ่งเช่น การเคลื่อนที่ การหมุน การวัดระยะทาง เป็นต้น ก็ให้ทำการกระทำนั้น แล้วทำการประมวลผลต่อไป ถ้าประมวลผลไปจนสุดต้นไม้แล้วให้ย้อนกลับไปเริ่มต้นที่รากของต้นไม้ใหม่จนกว่าจะสิ้นสุดเงื่อนไขในการวนซ้ำที่กำหนด

หลังจากทำการประมวลผลโปรแกรมแต่ละตัว แล้วใช้ฟังก์ชันวัดความเหมาะ ก็จะได้ค่าความเหมาะ หรือประสิทธิภาพในการแก้ปัญหาของโปรแกรมแต่ละตัว

2.4 <u>การสร้างประชากรของผลเฉลยรุ่นใหม่</u>

หลังจากที่ได้ประเมินค่าความเหมาะของประชากรทุกตัวแล้ว ขั้นตอนต่อไปเป็นการคัดเลือกประชากรที่มีประสิทธิภาพเพื่อให้อยู่รอดและขยายพันธุ์ต่อไป ตัวที่ไม่ถูกคัดเลือกก็จะสูญพันธุ์ไป แล้วทำการสร้างประชากรในรุ่นใหม่ขึ้นมา โดยมีแนวคิดว่า ประชากรในรุ่นใหม่ที่เกิดจากการนำประชากรที่ถูกคัดเลือกมาผสมพันธุ์กันจะมีประสิทธิภาพในการแก้ปัญหาสูงขึ้น วิธีการคัดเลือกประชากรมีอยู่ด้วยกันหลายวิธี แต่ในที่นี้จะกล่าวถึงวิธีคัดเลือกที่ใช้ในงานวิจัยนี้ คือทำการเรียงลำดับประชากรตามค่าความเหมาะของประชากรแต่ละตัว แล้วคัดเลือกโปรแกรมตามลำดับที่ได้เรียงไว้จนครบจำนวนที่ต้องการ

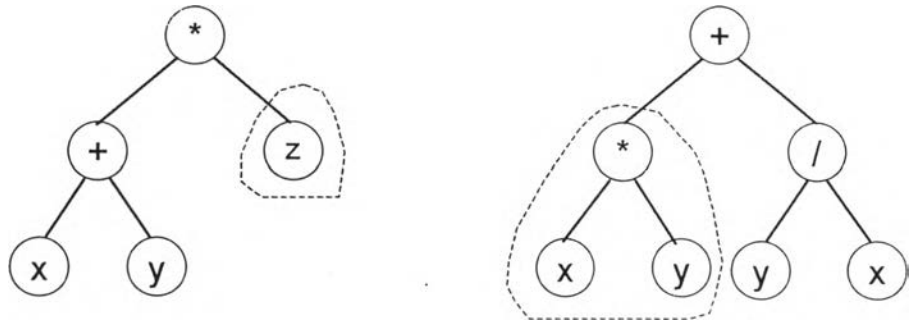
ขั้นตอนต่อไปเป็นการสร้างประชากรใหม่ในรุ่นต่อไป โดยอาศัยตัวดำเนินการทางพันธุกรรม (Genetic Operator) ตัวอย่างของตัวดำเนินการทางพันธุกรรมพื้นฐาน ได้แก่

2.4.1 *การสืบพันธุ์ (Reproduction)*

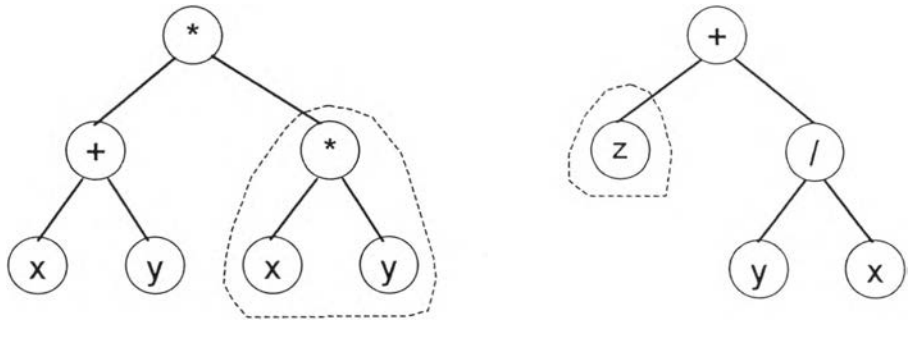
เป็นการทำสำเนา หรือคัดลอกประชากรที่ถูกคัดเลือกให้มีชีวิตอยู่ในรุ่นต่อไป โดยไม่มีการเปลี่ยน-แปลงใดๆ ในประชากรตัวนั้น เพื่อนำมาผสมพันธุ์สร้างโปรแกรมใหม่

2.4.2 *การไขว้เปลี่ยน (Crossover)*

เป็นการสร้างประชากรขึ้นมาใหม่ โดยนำประชากรที่ถูกคัดเลือก 2 ตัว มาผสมพันธุ์แลกเปลี่ยนบาง-ส่วนของโปรแกรมกัน โดยทำการสุ่มเลือกตำแหน่งที่ทำการแลกเปลี่ยน จะทำให้เกิดประชากรใหม่ขึ้นมา 2 ตัวที่มีลักษณะร่วมของพ่อและแม่ ดังตัวอย่างในรูปที่ 2.3 แสดงถึงประชากร 2 ตัวที่ถูกคัดเลือกเพื่อนำมาผสมพันธุ์กัน เส้นประแสดงตำแหน่งที่สุ่มเลือกที่จะทำการไขว้เปลี่ยน และรูปที่ 2.4 เป็นประชากรรุ่นลูกที่เกิดจากการผสมพันธุ์โดยตัวดำเนินการทางพันธุกรรมแบบไขว้เปลี่ยน



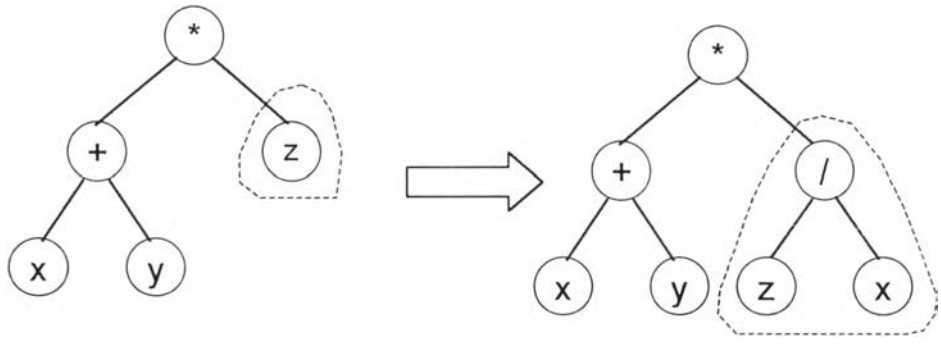
รูปที่ 2.3 โปรแกรม 2 ตัวที่ถูกคัดเลือกมาผสมพันธุ์กันโดยใช้ตัวดำเนินการไขว้เปลี่ยน



รูปที่ 2.4 โปรแกรม 2 ตัวที่เกิดขึ้นหลังจากการไขว้เปลี่ยน

2.4.3 การกลาย (Mutation)

เป็นการสร้างประชากรใหม่ขึ้นมา โดยทำการเปลี่ยนแปลงบางส่วนของประชากรที่ถูกคัดเลือก โดยสุ่มตำแหน่งที่จะทำการเปลี่ยนแปลง แล้วแทนที่ตำแหน่งนั้นด้วยโปรแกรมเล็กๆ ที่สร้างขึ้นอย่างสุ่ม ดังตัวอย่างในรูปที่ 2.5 แสดงโปรแกรมที่ถูกคัดเลือกเพื่อนำมาดำเนินการกลาย เส้นประแสดงตำแหน่งที่จะทำการเปลี่ยนแปลงและโปรแกรมที่เกิดขึ้นใหม่



รูปที่ 2.5 โปรแกรมที่ถูกคัดเลือกมาทำการกลาย (ซ้าย) และ โปรแกรมใหม่ที่เกิดหลังจากการกลาย (ขวา)

การสร้างประชากรของผลเฉลยในรุ่นใหม่ จะทำการสร้างจนได้จำนวนประชากรครบตามที่กำหนดไว้ จากนั้นก็ดำเนินการซ้ำในขั้นตอนการประเมินความเหมาะ และสร้างประชากรรุ่นใหม่ต่อไป ซึ่งในส่วนของการประเมินค่าความเหมาะ และการสร้างประชากรของผลเฉลยรุ่นใหม่นี้ จะเรียกว่าเป็น **กระบวนการวิวัฒนาการ** โดยเฉลี่ยแล้วประชากรในรุ่นใหม่ๆ จะมีค่าความเหมาะในการแก้ปัญหาสูงกว่าประชากรในรุ่นก่อน โดยปกติกระบวนการนี้จะสิ้นสุดลงเมื่อจำนวนรุ่นของประชากรครบตามจำนวนที่ได้กำหนดไว้ หรือจนกว่าจะได้ผลเฉลยที่ต้องการ โดยที่ผลเฉลยที่มีค่าความเหมาะที่ดีที่สุดที่ปรากฏในกระบวนการดังกล่าวจะถือเป็นคำตอบที่ได้จากกำหนดการเชิงพันธุกรรมสำหรับปัญหานั้นๆ

2.5 สรุปท้ายบท

โดยสรุปแล้ว กำหนดการเชิงพันธุกรรมเป็นวิธีการเรียนรู้และค้นหาคำตอบ โดยเลียนแบบจากกระบวนการวิวัฒนาการในธรรมชาติ โดยการแทนคำตอบให้อยู่ในรูปโปรแกรมคอมพิวเตอร์ โดยมีขั้นตอนเริ่มต้นจากการสร้างกลุ่มคำตอบที่เป็นไปได้ขึ้นมาเป็นจำนวนมาก แล้วทำการวิวัฒนาการคำตอบเหล่านั้นจนได้เป็นคำตอบที่แท้จริง ที่สามารถแก้ไขปัญหาที่กำหนดได้ กระบวนการวิวัฒนาการคำตอบเป็นขั้นตอนที่มีความสำคัญต่อคำตอบที่ถูกสร้างขึ้นมา

การนำวิธีกำหนดการเชิงพันธุกรรมไปใช้ในการแก้ปัญหา ต้องคำนึงถึงปัจจัยต่างๆ และควรเลือกตัวแปรต่างๆ ให้เหมาะสมกับปัญหา ตัวแปรและปัจจัยในที่นี้ ได้แก่ การกำหนดจำนวนประชากรในแต่ละรุ่น จำนวนรุ่น ขนาดของต้นไม้ที่แทนประชากร การเลือกใช้ตัวดำเนินการทางพันธุกรรม จำนวนประชากรของผลเฉลยที่เกิดจากการไขว้เปลี่ยน และการกลาย วิธีการเลือกประชากรเพื่อนำมาสืบพันธุ์ รวมไปถึงการออกแบบฟังก์ชัน เทอร์มินอล และฟังก์ชันค่าความเหมาะที่ดีด้วย