บทที่ 1

บทนำ

1.1 ความสำคัญและความเป็นมาของปัญหา

กำหนดการเชิงพันธุกรรม(genetic programming) เป็นวิธีการหนึ่งที่ใช้สำหรับการเรียนรู้ (learning) ซึ่งเป็นวิธีการค้นหาคำตอบโดยใช้การจำลองแบบตามกฎการคัดเลือกโดยธรรมชาติ(natural selection) ประกอบกับการใช้ตัวปฏิบัติการทางพันธุกรรม(genetic operator) ต่างๆ กำหนดการเชิงพันธุกรรมเป็นวิธีการที่มีประสิทธิภาพสำหรับปัญหาหลายประเภท แต่เป็นวิธีการที่ต้องใช้การคำนวณสูง วิธีการหนึ่งสำหรับวัดประสิทธิภาพของวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรมคือการวัดด้วยค่าของ *ความเพียรพยายามเชิงคำนวณ* (computational effort) ซึ่งเป็นค่าที่ได้จากการคำนวณทางสถิติเพื่อหาจำนวนผลเฉลย(solution) อย่างน้อยที่ต้องประมวลผลจนพบคำตอบภายใต้ความมั่นใจที่กำหนด ในการเปรียบเทียบประสิทธิภาพของวิธีการเรียนรู้ วิธีการเรียนรู้ที่คำนวณค่าของความเพียรพยายามเชิงคำนวณได้ต่ำกว่าสำหรับการหาคำตอบของปัญหาเดียวกัน จะแสดงถึงความมีประสิทธิภาพในการเรียนรู้ที่สูงกว่า

จุมพล พลวิชัย (2539) ได้ทำการวิจัยในหัวข้อ *การเรียนรู้ของแขนหุ่นยนต์โดยใช้การโปรแกรมพันธุการ* โดยมีจุดประสงค์เพื่อให้สามารถสร้างโปรแกรมหุ่นยนต์โดยใช้กำหนดการเชิงพันธุกรรมให้สามารถเคลื่อนแขนหุ่นยนต์ไปยังเป้าหมายโดยต้องหลีกเลี่ยงสิ่งกีดขวาง โดยทำการทดลองทั้งสิ้น 16 การทดลอง ข้อมูลนำเข้าที่จำเป็นต้องใช้คือภาพที่ได้มาจากการถ่ายโดยกล้องวีดีทัศน์ ผลการวิจัยพบว่าสามารถสร้างโปรแกรมหุ่นยนต์ที่สามารถแก้ปัญหาสิ่งกีดขวางทั้ง 16 การทดลองได้ แต่มีค่าของความเพียรพยายามเชิงคำนวณสูงและไม่เท่ากันในแต่ละการทดลอง ซึ่งค่าของความเพียรพยายามเชิงคำนวณที่สูงนี้แสดงถึงความมีประสิทธิภาพในการเรียนรู้ที่ต่ำ

การวิจัยในครั้งนี้ได้รับแรงบันดาลใจมาจากการวิจัย *การเรียนรู้ของแขนหุ่นยนต์โดยใช้การโปรแกรมพันธุการ* ดังกล่าว เพื่อเป็นกรณีศึกษาในการลดทอนความเพียรพยายามเชิงคำนวณของวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรม เนื่องจากมีความพร้อมของซอฟต์แวร์

และข้อมูลของการวิจัยเดิม ผลของการวิจัยครั้งนี้จะทำให้ได้แนวทางในการลดทอนความเพียรพยายามเชิงคำนวณของวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรมแก่ระบบอื่นๆต่อไป

1.2 **วัตถุประสงค์**

เพื่อเสนอแนวทางในการลดทอนความเพียรพยายามเชิงคำนวณของวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรม

1.3 **ขอบเขตของการวิจัย**

1. ใช้การวิจัย *การเรียนรู้ของแขนหุ่นยนต์โดยใช้การโปรแกรมพันธุการ* (จุมพล พลวิชัย, 2539) เป็นกรณีศึกษาในการลดทอนความเพียรพยายามเชิงคำนวณของวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรม
2. ไม่ทำการทดลองกับแขนหุ่นยนต์จริง

1.4 **ประโยชน์ที่คาดว่าจะได้รับ**

ได้แนวทางในการลดทอนความเพียรพยายามเชิงคำนวณของวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรมซึ่งสามารถนำไปใช้กับระบบอื่นๆได้

1.5 **การวิจัยที่เกี่ยวข้อง**

1.5.1 **การวิจัยปัญหาแขนหุ่นยนต์**

จุมพล พลวิชัย (2539) ได้ทำการวิจัยในหัวข้อ *การเรียนรู้ของแขนหุ่นยนต์โดยใช้การโปรแกรมพันธุการ* เป็นการศึกษาการใช้กระบวนการกำหนดการเชิงพันธุกรรมเพื่อสร้างโปรแกรมหุ่นยนต์ให้สามารถเคลื่อนแขนหุ่นยนต์ไปยังเป้าหมายโดยต้องหลีกเลี่ยงสิ่งกีดขวางได้ ข้อมูลนำเข้าที่จำเป็นต้องใช้คือภาพที่ได้มาจากการถ่ายโดยกล้องวีดีทัศน์ ซึ่งจะบันทึกสภาพแวดล้อมได้แก่ ตำแหน่งของแขนหุ่นยนต์ สิ่งกีดขวาง และเป้าหมาย เพื่อใช้นำทางแขนหุ่นยนต์ไปสู่เป้าหมาย

ในการทำการทดลองใช้การจำลองแบบโดยกำหนดการทดลองออกเป็น 4 กลุ่มตามสภาพแวดล้อมของแขนหุ่นยนต์ได้แก่ กลุ่มของสภาพแวดล้อมที่ไม่มีสิ่งกีดขวาง กลุ่มของสภาพแวดล้อมที่มีสิ่งกีดขวาง 1 สิ่งกีดขวาง กลุ่มของสภาพแวดล้อมที่มีสิ่งกีดขวางมากกว่า 1 สิ่งกีดขวาง และกลุ่มของสภาพแวดล้อมที่มีลักษณะของปัญหาจุดต่ำสุดเสมือน ในแต่ละกลุ่มการทดลองยังแบ่งเป็นอีก 4 การทดลองย่อย

ผลการวิจัยพบว่าสามารถสร้างโปรแกรมหุ่นยนต์ที่สามารถแก้ปัญหาสิ่งกีดขวางทั้ง 16 การทดลองได้ โดยผลการทดลองในสภาพแวดล้อมที่ไม่ซับซ้อนจะคำนวณค่าของความเพียรพยายามเชิงคำนวณได้ต่ำ และคำนวณค่าของความเพียรพยายามเชิงคำนวณได้สูงขึ้นสำหรับผลการทดลองในสภาพแวดล้อมที่ซับซ้อนมากขึ้นตามลำดับ เมื่อนำผลการทดลองมาทดสอบกับสภาพแวดล้อมจริง พบว่าโปรแกรมหุ่นยนต์ 83 เปอร์เซนต์สามารถทำงานได้ การที่โปรแกรมควบคุมแขนหุ่นยนต์ไม่สามารถทำงานได้ทั้งหมดเป็นผลมาจากความแตกต่างกันของการทำงานในสภาพแวดล้อมในการจำลองแบบและสภาพแวดล้อมจริง

Prabhas Chongstitvatana และ Jumpol Polvichai (1996) ได้ทำรายงานการวิจัยเรื่อง *Learning a Visual Task by Genetic Programming* ซึ่งเป็นการวิจัยปัญหาแขนหุ่นยนต์ต่อจากการวิจัยของ จุมพล พลวิชัย (2539) โดยทำการทดลองกับสภาพแวดล้อม 3 สภาพแวดล้อมซึ่งคัดเลือกมาจากการวิจัยเดิม ผลการวิจัยพบว่าสามารถสร้างโปรแกรมหุ่นยนต์ที่สามารถแก้ปัญหาสิ่งกีดขวางทั้ง 3 สภาพแวดล้อมได้ เมื่อนำผลการทดลองมาทดสอบกับสภาพแวดล้อมจริงพบว่า 2 ใน 3 สภาพแวดล้อมโปรแกรมหุ่นยนต์ในการวิจัยนี้มีเปอร์เซนต์ของความสำเร็จสูงกว่าในการวิจัยของ จุมพล พลวิชัย (2539)

1.5.2 การวิจัยเพื่อเพิ่มประสิทธิภาพวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรม

การวิจัยเพื่อเพิ่มประสิทธิภาพวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรมนี้ ได้มีการวิจัยในหลายแนวทางดังจะยกตัวอย่างพอเป็นสังเขป 2 ตัวอย่างดังนี้

1. **ฟังก์ชันซึ่งถูกนิยามโดยอัตโนมัติ** (automatically defined functions, เอดีเอฟ) เอดีเอฟเป็นวิธีที่ได้รับการเสนอโดย Koza (1992) โดยวิธีเอดีเอฟนี้โครงสร้างของแต่ละผลเฉลยที่ถูกสร้างในกระบวนการเรียนรู้แบบกำหนดการเชิงพันธุกรรมจะประกอบไปด้วยฟังก์ชันหลักและฟังก์ชันย่อย ซึ่งต่างจากกำหนดการเชิงพันธุกรรมปกติซึ่งโครงสร้างของแต่ละผลเฉลยมีเพียงฟังก์ชันหลักเพียงฟังก์ชันเดียว โครงสร้างของผลเฉลยของวิธีเอดีเอฟจะขึ้นอยู่กับ

พารามิเตอร์(parameter)ต่างๆ ได้แก่ จำนวนฟังก์ชัน จำนวนอาร์กิวเมนต์(argument)ของแต่ละฟังก์ชัน และการอ้างอิงระหว่างฟังก์ชันซึ่งต้องมีการกำหนดก่อนหน้า การเลือกใช้พารามิเตอร์ต่างๆนี้จำเป็นต้องมีการศึกษาปัญหามาเป็นอย่างดี เนื่องจากปัญหาแต่ละปัญหาจะมีพารามิเตอร์ที่เหมาะสมแตกต่างกัน

2. **การแสวงหามอดูล** (module acquisition, เอ็มเอ) เอ็มเอเป็นวิธีที่ได้รับการเสนอโดย Angeline (1994) วิธีการนี้ในกระบวนการเรียนรู้จะมีการสุ่มเลือกผลเฉลยเพื่อทำการตัดส่วนและสร้างเป็นฟังก์ชันย่อย ฟังก์ชันย่อยที่ถูกสร้างด้วยกระบวนการนี้จะถูกเก็บไว้ในคลัง (library) ซึ่งสามารถถูกเรียกใช้โดยผลเฉลยอื่นๆได้ ฟังก์ชันในคลังจะถูกเก็บรักษาไว้จนกว่าจะไม่มีผลเฉลยใดเรียกใช้ ฟังก์ชันย่อยจะถูกเรียกใช้จากผลเฉลยเพียงผลเฉลยเดียวเมื่อถูกสร้างขึ้นในครั้งแรก แต่การเรียกใช้ในกระบวนการเรียนรู้จะเพิ่มมากขึ้นหากพบว่าฟังก์ชันย่อยนั้นเป็นฟังก์ชันที่ดี

1.6 <u>ทฤษฎีที่เกี่ยวข้อง</u>

1.6.1 กำหนดการเชิงพันธุกรรม

กำหนดการเชิงพันธุกรรมเป็นวิธีการหนึ่งที่ใช้สำหรับการเรียนรู้ ซึ่งเป็นวิธีการที่ได้รับการพัฒนามาจากขั้นตอนวิธีเชิงพันธุกรรม(genetic algorithm) ซึ่งคิดค้นโดย Holland (1975) ขั้นตอนวิธีเชิงพันธุกรรมเป็นกระบวนการค้นหาคำตอบโดยการจำลองแบบตามกฎการคัดเลือกโดยธรรมชาติ ซึ่งมีลักษณะที่แตกต่างไปจากเทคนิคอื่นทางปัญญาประดิษฐ์ (artificial intelligence) เนื่องจากเทคนิคทางปัญญาประดิษฐ์ส่วนใหญ่ในการแก้ปัญหาหนึ่งๆ จะเริ่มจากต้องสร้างความรู้(knowledge) ในการแก้ปัญหานั้นเสียก่อน จากนั้นจึงทำการค้นหาคำตอบโดยอ้างอิงกับความรู้นั้นๆ ส่วนขั้นตอนวิธีเชิงพันธุกรรมไม่ใช้ความรู้ในการค้นหาคำตอบ โดยในการแก้ปัญหาจะเริ่มจากการสร้างกลุ่มของคำตอบที่เป็นไปได้เป็นขั้นตอนแรกและผลลัพธ์ที่ได้จะมีลักษณะเป็นกลุ่มของคำตอบ

ข้อแตกต่างระหว่างกำหนดการเชิงพันธุกรรมและขั้นตอนวิธีเชิงพันธุกรรมคือลักษณะของการแทนคำตอบ กำหนดการเชิงพันธุกรรมจะแทนคำตอบในลักษณะของโปรแกรมคอมพิวเตอร์ ส่วนขั้นตอนวิธีเชิงพันธุกรรมจะแทนลักษณะของคำตอบเป็นสตริงก์ขนาดคงที่

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การค้นหาคำตอบโดยกำหนดการเชิงพันธุกรรม อาจแบ่งออกเป็นขั้นตอนหลักคือ การสร้างประชากรของผลเฉลยเริ่มต้น การตรวจสอบประสิทธิภาพของผลเฉลย การสร้างประชากรของผลเฉลยรุ่นใหม่ และการหาผลลัพธ์

1. **การสร้างประชากรของผลเฉลยเริ่มต้น** ขั้นตอนนี้เป็นขั้นตอนเริ่มต้นของกำหนดการเชิงพันธุกรรมซึ่งจะทำการสร้างประชากรของผลเฉลยแบบสุ่ม(random) ในแต่ละผลเฉลยจะประกอบไปด้วยส่วนประกอบ 2 ชนิดคือ ฟังก์ชัน(function) และเทอมินอล (terminal)

ฟังก์ชัน ฟังก์ชันที่ใช้ในผลเฉลยอาจเป็นฟังก์ชันปฏิบัติการพื้นฐาน (เช่น การบวก การลบ) ฟังก์ชันมาตรฐานทางคณิตศาสตร์ (เช่น sin log) ฟังก์ชันมาตรฐานในการโปรแกรม (เช่น if-then-else) หรือฟังก์ชันทางตรรก (เช่น and or) เป็นต้น

เทอมินอล โดยปกติจะเป็นเซตของปัจจัยที่เป็นอิสระแก่กันสำหรับปัญหานั้นๆ เช่น ปัญหาการหาทางออกจากทางวงกต เทอมินอลคือคำสั่งในการ เลี้ยวซ้าย เลี้ยวขวา และตรงไปเป็นต้น ในกรณีอื่นๆ เทอมินอลอาจหมายถึงฟังก์ชันที่ไม่มีอาร์กิวเมนต์

ผลลัพธ์จากขั้นตอนนี้จะได้ประชากรของผลเฉลยจำนวนมาก ผลเฉลยแต่ละผลเฉลยจะมีจำนวนและชนิดของฟังก์ชันและเทอมินอลต่างๆกันอันเป็นผลมาจากการสุ่ม ซึ่งแต่ละผลเฉลยจะถูกนำไปตรวจสอบประสิทธิภาพในขั้นตอนต่อไป ขั้นตอนนี้อาจเปรียบเทียบได้กับการก่อกำเนิดสิ่งมีชีวิตในทางพันธุศาสตร์ ผลเฉลยแต่ละผลเฉลยเปรียบเสมือนโครโมโซม (chromosome) โดยฟังก์ชันและเทอมินอลเปรียบเสมือนจีน(gene) บนโครโมโซมนั้นๆ สำหรับกำหนดการเชิงพันธุกรรมคำว่า ผลเฉลย(solution), โปรแกรมคอมพิวเตอร์(computer program) และโครโมโซม(chromosome) จะมีความหมายเดียวกัน

2. **การตรวจสอบประสิทธิภาพของผลเฉลย** ในขั้นตอนนี้จะทำการตรวจสอบประสิทธิภาพของผลเฉลยแต่ละผลเฉลย โดยจะมีผลเฉลยส่วนน้อยเท่านั้นที่มีประสิทธิภาพดี ซึ่งจะนำไปสร้างประชากรของผลเฉลยในรุ่น(generation) ใหม่ต่อไป ซึ่งเป็นผลมาจากสมมติฐานที่ว่าโครโมโซมส่วนใหญ่จะไม่มีประสิทธิภาพซึ่งจะถูกขจัดทิ้ง มีเพียงโครโมโซมส่วนน้อยที่ได้รับการคัดเลือกเพื่อสร้างเป็นโครโมโซมรุ่นใหม่ต่อไป การตรวจสอบประสิทธิภาพของผลเฉลยนี้จะตรวจวัดโดยการคำนวณค่าความเหมาะ(fitness value) โดยใช้ฟังก์ชันความเหมาะ(fitness function) ในกรณีที่ค่าความเหมาะสูงแสดงถึงประสิทธิภาพดี กลุ่มของผล

เฉลยที่มีค่าความเหมาะสูงจะมีโอกาสถูกนำไปสร้างประชากรของผลเฉลยในรุ่นใหม่ต่อไปมากกว่ากลุ่มที่ค่าความเหมาะต่ำ

การนิยามฟังก์ชันความเหมาะเป็นส่วนที่ยากและสำคัญที่สุดของแนวคิดกำหนดการเชิงพันธุกรรม ฟังก์ชันความเหมาะจะแสดงให้เห็นถึงความมีประสิทธิภาพของผลเฉลยที่จะแก้ปัญหาจากโจทย์นั้นๆ ฟังก์ชันความเหมาะของแต่ละปัญหาจะมีความแตกต่างกัน ตัวอย่างเช่นหากใช้กำหนดการเชิงพันธุกรรมในการตั้งเวลาของนาฬิกา ในกรณีนี้ฟังก์ชันความเหมาะมีลักษณะง่ายคืออาจวัดเป็นปริมาณของเวลาที่ผิดพลาด แต่ในทางปฏิบัติแล้วมีปัญหาส่วนน้อยเท่านั้นที่ฟังก์ชันความเหมาะมีลักษณะที่ง่าย

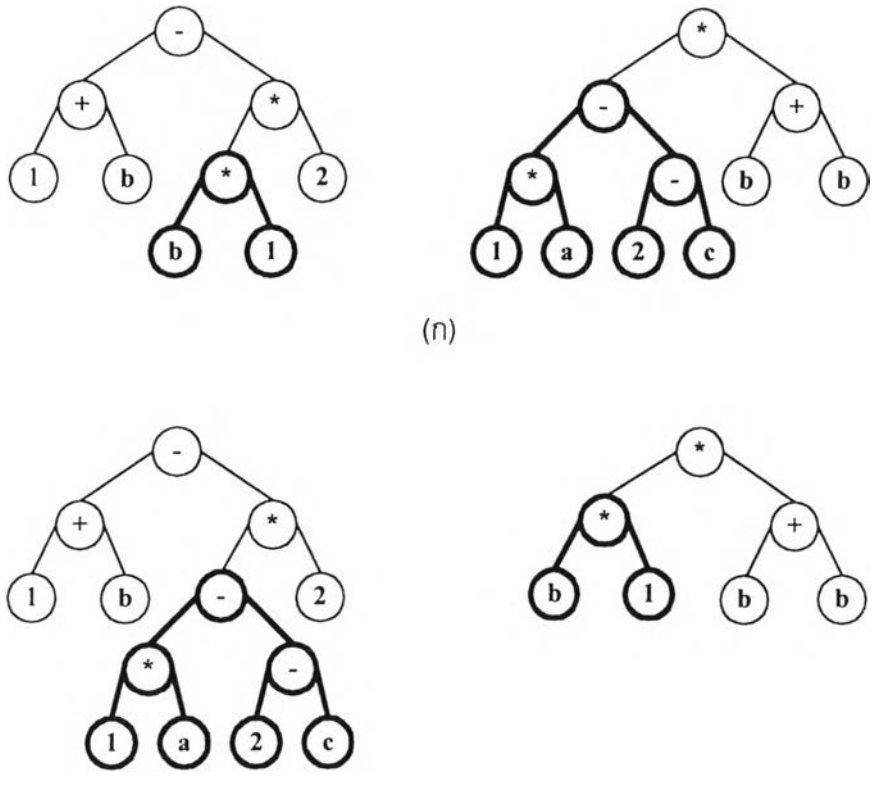
3. **การสร้างประชากรของผลเฉลยรุ่นใหม่** ขั้นตอนนี้จะเป็นการสร้างประชากรของผลเฉลยรุ่นใหม่โดยการสืบพันธุ์(reproduction) การไขว้เปลี่ยน(crossover) และการกลาย(mutation)

การสืบพันธุ์ เป็นการสำเนากลุ่มของผลเฉลยที่มีประสิทธิภาพดีที่สุดของรุ่นเดิมมาเป็นประชากรของผลเฉลยรุ่นใหม่ สำหรับผลเฉลยที่มีประสิทธิภาพต่ำจะตกการพิจารณาซึ่งเปรียบเทียบได้กับกฎการคัดเลือกโดยธรรมชาติ

การไขว้เปลี่ยน เป็นกระบวนการที่สำคัญที่สุดกระบวนการหนึ่งของกำหนดการเชิงพันธุกรรม กระบวนการนี้จะทำการสุ่มผลเฉลยรุ่นเดิมครั้งละ 2 ผลเฉลย(parents) มาทำการสร้างเป็นผลเฉลยรุ่นใหม่ 2 ผลเฉลย(children หรือ offspring) การสุ่มเลือกจะมีผลมาจากค่าความเหมาะของแต่ละผลเฉลย ทำให้โอกาสที่จะถูกเลือกของแต่ละผลเฉลยไม่เท่ากัน โดยผลเฉลยที่มีค่าความเหมาะสูงกว่าจะมีโอกาสได้รับเลือกมากกว่า ตัวอย่างของการไขว้เปลี่ยนแสดงดังรูปที่ 1.1

การกลาย เป็นกระบวนการที่สำคัญอีกกระบวนการหนึ่งซึ่งเปรียบเทียบได้กับการกลายในทางพันธุศาสตร์ ในกระบวนการนี้จะทำการสุ่มเลือกผลเฉลยรุ่นเดิมครั้งละ 1 ผลเฉลย แต่ละผลเฉลยมีโอกาสถูกเลือกไม่เท่ากัน โดยผลเฉลยที่มีค่าความเหมาะสูงกว่าจะมีโอกาสได้รับเลือกมากกว่า จากนั้นจึงทำการเปลี่ยนแปลงผลเฉลยนั้นโดยการสุ่มตำแหน่งที่จะเปลี่ยนแปลงตัวอย่างของการกลายแสดงดังรูปที่ 1.2

จากนั้นประชากรของผลเฉลยรุ่นใหม่ที่ได้ในขั้นตอนนี้ จะถูกดำเนินการซ้ำในขั้นตอนตรวจสอบประสิทธิภาพใหม่ และนำไปสร้างเป็นประชากรผลเฉลยในรุ่นต่อๆไป



รูปที่ 1.1 แสดงตัวอย่างการไขว้เปลี่ยนจากผลเฉลยรุ่นเดิม 2 ผลเฉลย (ก) เป็นผลเฉลยรุ่นใหม่ 2 ผลเฉลย (ข)

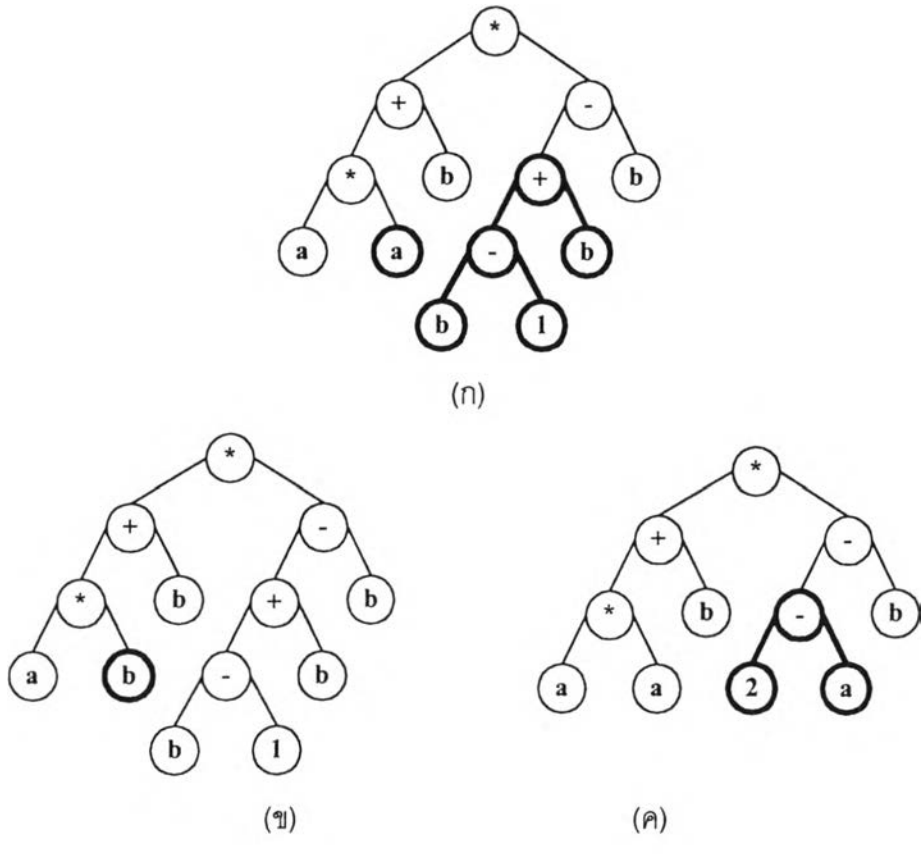
4. **การหาผลลัพธ์** ขั้นตอนนี้เป็นการหาผลลัพธ์ของกำหนดการเชิงพันธุกรรม โดยผลเฉลยหรือกลุ่มของผลเฉลยที่มีประสิทธิภาพดีที่สุด(ประเมินด้วยฟังก์ชันความเหมาะ)ที่ปรากฏในขั้นตอนทั้งหมดที่กล่าวมาข้างต้นจะเป็นผลลัพธ์ของกำหนดการเชิงพันธุกรรม

การใช้เทคนิค(technique)ของกำหนดการเชิงพันธุกรรมนอกจากการกำหนด ฟังก์ชันเทอมินอล และฟังก์ชันความเหมาะแล้วยังต้องกำหนดตัวแปรอื่นซึ่งแบ่งได้เป็น 2 กลุ่มคือ พารามิเตอร์ที่ใช้ควบคุมในการทำงาน และเงื่อนไขในการสิ้นสุดการทำงาน

1. **พารามิเตอร์ที่ใช้ควบคุมในการทำงาน** ตัวแปรในกลุ่มนี้ได้แก่จำนวนของประชากรของผลเฉลย จำนวนประชากรจะมีผลต่อความเร็วในการค้นหาคำตอบ และการใช้

ทรัพยากรของระบบ นอกจากจำนวนประชากรแล้วตัวแปรอื่นๆได้แก่ จำนวนรุ่น (number of generation) อัตราการสืบพันธุ์ อัตราการการผสมพันธุ์ และอัตราการกลายเป็นต้น

2. **เงื่อนไขในการสิ้นสุดการทำงาน** โดยปกติจะกำหนดให้การทำงานสิ้นสุดเมื่อครบจำนวนรุ่น



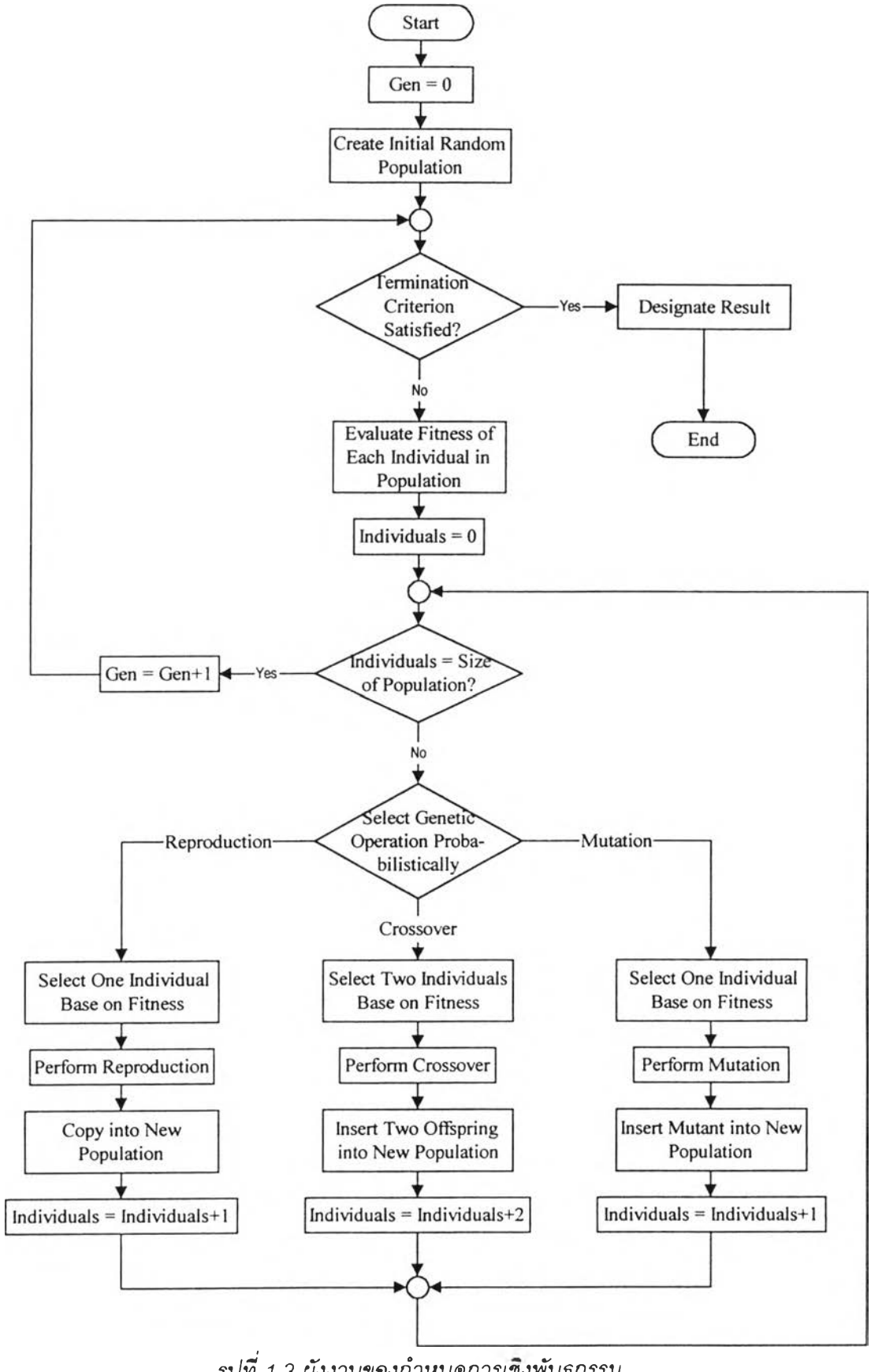
รูปที่ 1.2 แสดงตัวอย่างการกลาย

(ก) ผลเฉลยรุ่นเดิมที่ถูกเลือก

(ข) ผลเฉลยรุ่นใหม่ซึ่งเกิดการกลายโดยการเปลี่ยนเทอมินอล

(ค) ผลเฉลยรุ่นใหม่ซึ่งเกิดการกลายโดยการเปลี่ยนต้นไม้ส่วนย่อย(subtree)

โดยสรุป กำหนดการเชิงพันธุกรรมเป็นเทคนิคที่พัฒนามาจากขั้นตอนวิธีเชิงพันธุกรรม ซึ่งทำงานได้ดีสำหรับปัญหาหลายประเภทเช่น ปัญหาที่มีตัวแปรจำนวนมาก และปัญหาที่ไม่มีคำตอบที่สมบูรณ์ (no ideal solution) กำหนดการเชิงพันธุกรรมเป็นเทคนิคที่ง่ายต่อการทำงานแบบขนาน (easily parallelized) แต่เป็นเทคนิคที่ต้องใช้การคำนวณสูง (computationally expensive) ขั้นตอนต่างๆของกำหนดการเชิงพันธุกรรมสามารถดังรูปที่ 1.3



รูปที่ 1.3 ผังงานของกำหนดการเชิงพันธุกรรม

1.6.2 ความเพียรพยายามเชิงคำนวณ

เนื่องจากกระบวนการของกำหนดการเชิงพันธุกรรมเป็นกระบวนการที่ใช้การสุ่ม ซึ่งทำให้ในแต่ละครั้งของการทดลองได้ผลลัพธ์ที่ไม่เหมือนกัน การวัดประสิทธิภาพของวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรมจึงต้องใช้การคำนวณทางสถิติโดยใช้ข้อมูลจากผลของการทดลองซ้ำหลายๆครั้ง โดยในที่นี้จะใช้คำว่า *รอบของการทดลอง* แทนการทดลองซ้ำในแต่ละครั้ง

ตามหนังสือ Genetic Programming II ของ Koza (1994) ได้กล่าวถึงความเพียรพยายามเชิงคำนวณซึ่งเป็นวิธีหนึ่งในการวัดประสิทธิภาพของวิธีการเรียนรู้แบบกำหนดการเชิงพันธุกรรมโดยใช้การคำนวณทางสถิติเพื่อคำนวณหาจำนวนผลเฉลยอย่างน้อยที่ต้องประมวลผลจนพบผลเฉลยที่ประสบความสำเร็จ(หมายถึงผลเฉลยที่มีประสิทธิภาพซึ่งยอมรับว่าเป็นคำตอบของปัญหา)ภายใต้ความมั่นใจที่กำหนดดังสรุปได้ดังนี้

ในการดำเนินการทดลองของกำหนดการเชิงพันธุกรรมซึ่งมีจำนวนประชากรของผลเฉลยในแต่ละรุ่นคงที่เท่ากับ M และมีจำนวนรุ่นเท่ากับ G หากสามารถพบผลเฉลยที่ประสบความสำเร็จในรุ่น (generation) เดียวกันทุกรอบของการทดลอง จำนวนผลเฉลยอย่างน้อยที่ต้องประมวลผลจนพบผลเฉลยที่ประสบความสำเร็จจะมีค่าเท่ากับผลคูณของจำนวนประชากร M กับจำนวนรุ่นที่ใช้ในการหาผลเฉลยที่ประสบความสำเร็จ หากไม่พบผลเฉลยที่ประสบความสำเร็จในรุ่นเดียวกันในแต่ละรอบของการทดลอง แต่สามารถพบผลเฉลยที่ประสบความสำเร็จภายใน G รุ่นทุกรอบของการทดลอง จำนวนผลเฉลยอย่างน้อยที่ต้องประมวลผลจนพบผลเฉลยที่ประสบความสำเร็จจะมีค่าเท่ากับผลคูณของจำนวนประชากร M กับค่าเฉลี่ยของจำนวนรุ่นที่ใช้จนพบผลเฉลยที่ประสบความสำเร็จในแต่ละรอบของการทดลอง แต่เนื่องจากกำหนดการเชิงพันธุกรรมเป็นขั้นตอนวิธีที่ใช้หลักของความน่าจะเป็นซึ่งไม่สามารถรับประกันได้ว่าในการทดลองทุกรอบจะสามารถพบผลเฉลยที่ประสบความสำเร็จได้ภายในจำนวนรุ่นที่กำหนด จึงต้องใช้การคำนวณทางสถิติเพื่อหาจำนวนผลเฉลยอย่างน้อยที่ต้องประมวลผลจนพบผลเฉลยที่ประสบความสำเร็จภาพใต้ความมั่นใจที่กำหนด

หากกำหนดให้ $Y(M,i)$ แทนความน่าจะเป็นที่จะพบผลเฉลยที่ประสบความสำเร็จในรุ่นที่ i ซึ่งคำนวณได้จากจำนวนรอบของการทดลองที่พบผลเฉลยที่ประสบความสำเร็จในรุ่นที่ i หารด้วยจำนวนรอบของการทดลองทั้งหมด จะสามารถคำนวณหาค่าความน่าจะเป็นสะสมของความสำเร็จ $P(M,i)$ ได้ โดย $P(M,i)$ จะเป็นฟังก์ชันที่ไม่ลดลง หากในทุกรอบ

ของการทดลองพบผลเฉลยที่ประสบความสำเร็จภายใน G รุ่นดังกล่าวมาข้างต้น จะได้ว่า $P(M,G)$ จะมีค่าเท่ากับ 1.0 แต่ในความเป็นจริงจะพบว่า $P(M,G)$ มักจะมีค่าน้อยกว่า 1.0 เนื่องจากจะมีบางรอบของการทดลองที่ไม่สามารถพบผลเฉลยที่ประสบความสำเร็จได้ภายใน G รุ่น

หากกำหนดให้ $R(M,i,z)$ แทนจำนวนรอบของการทดลองเพื่อให้สามารถพบผลเฉลยที่ประสบความสำเร็จภายในรุ่นที่ i โดยมีความมั่นใจของความสำเร็จเท่ากับ z (เช่น z เท่ากับ 99 เปอร์เซนต์) จะได้ว่า

$$z = 1 - \left[1 - P(M,i)\right]^{R(M,i,z)}$$

ซึ่งสามารถหา $R(M,i,z)$ ได้จากสมการ

$$R(M,i,z) = \left\lceil \frac{\log(1-z)}{\log(1-P(M,i))} \right\rceil$$

หากกำหนดให้ $I(M,i,z)$ แทนจำนวนผลเฉลยอย่างน้อยที่ต้องประมวลผลจนพบผลเฉลยที่ประสบความสำเร็จภายในรุ่นที่ i โดยมีค่าความมั่นใจของความสำเร็จ z จะได้ว่า

$$I(M,i,z) = M \times i \times R(M,i,z)$$

สำหรับรุ่นที่มีค่า $I(M,i,z)$ น้อยที่สุดเรียกว่ารุ่นที่ดีที่สุด (best generation, i^*) ซึ่งค่า $I(M,i,z)$ ของรุ่นที่ดีที่สุดนี้เรียกว่า *ความเพียรพยายามเชิงคำนวณ* (E) โดย

$$E = I(M,i^*,z) = M \times i^* \times R(M,i^*,z)$$

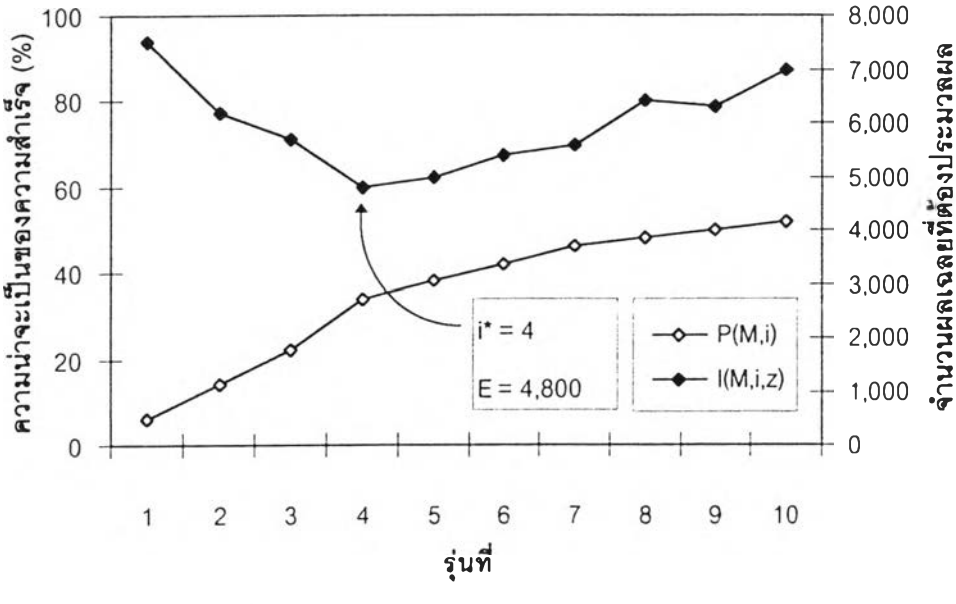
ซึ่งค่าความเพียรพยายามเชิงคำนวณที่คำนวณได้นี้ไม่จำเป็นจะต้องเป็นค่าความเพียรพยายามเชิงคำนวณน้อยที่สุดจริงๆของปัญหานั้น

ตัวอย่างการคำนวณค่าความเพียรพยายามเชิงคำนวณของตัวอย่างปัญหาแสดงดังตารางที่ 1.1 โดยกำหนดจำนวนประชากรของผลเฉลยในแต่ละรุ่นเท่ากับ 100 จำนวนรุ่นเท่ากับ 10 รุ่น จำนวนรอบของการทดลองเท่ากับ 50 รอบ ค่าความมั่นใจของความสำเร็จเท่ากับ 99 เปอร์เซนต์ ซึ่งพบว่าค่าความเพียรพยายามเชิงคำนวณของตัวอย่างปัญหานี้เท่ากับ 4,800 โดยรุ่นที่ดีที่สุดคือรุ่นที่ 4

กราฟที่แสดงความน่าจะเป็นสะสมของความสำเร็จ $P(M,i)$ และจำนวนผลเฉลยอย่างน้อยที่ต้องประมวลผลจนพบผลเฉลยที่ประสบความสำเร็จ $I(M,i,z)$ ในแต่ละรุ่น i เรียกว่า *เส้นโค้งสมรรถนะ*(performance curve) สำหรับเส้นโค้งสมรรถนะของตัวอย่างปัญหาแสดงดังรูปที่ 1.4

ตารางที่ 1.1 ค่าของตัวแปรต่างๆในการหาค่าความเพียรพยายามเชิงคำนวณของตัวอย่างปัญหา

รุ่นที่	จำนวนรอบของการทดลองที่พบผลเฉลยที่ประสบความสำเร็จ	$Y(M,i)$	$P(M,i)$	$R(M,i,z)$	$I(M,i,z)$
1	3	0.06	0.06	75	7,500
2	4	0.08	0.14	31	6,200
3	4	0.08	0.22	19	5,700
4	6	0.12	0.34	12	4,800
5	2	0.04	0.38	10	5,000
6	2	0.04	0.42	9	5,400
7	2	0.04	0.46	8	5,600
8	1	0.02	0.48	8	6,400
9	1	0.02	0.05	7	6,300
10	1	0.02	0.52	7	7,000



รูปที่ 1.4 เส้นโค้งสมรรถนะของตัวอย่างปัญหา

ในการเปรียบเทียบโดยใช้ค่าความเพียรพยายามเชิงคำนวณ หากใช้วิธีในการเรียนรู้เดียวกันในการหาคำตอบของปัญหาที่ต่างกัน ปัญหาที่มีค่าความเพียรพยายามเชิงคำนวณสูงกว่าจะแสดงถึงปัญหานั้นมีความยากกว่าปัญหาที่มีค่าความเพียรพยายามเชิงคำนวณต่ำกว่า และหากใช้วิธีในการเรียนรู้ต่างกันในการหาคำตอบของปัญหาเดียวกัน วิธีการเรียนรู้ที่คำนวณค่าความเพียรพยายามเชิงคำนวณได้ต่ำกว่าแสดงถึงความมีประสิทธิภาพในการเรียนรู้มากกว่าวิธีการเรียนรู้ที่คำนวณค่าความเพียรพยายามเชิงคำนวณได้สูงกว่า

1.6.3 ฟังก์ชันซึ่งถูกนิยามโดยอัตโนมัติ

ในกระบวนการกำหนดการเชิงพันธุกรรมปกติ ลักษณะของผลเฉลยที่ถูกสร้างขึ้นหากเปรียบเทียบกับโปรแกรมคอมพิวเตอร์ที่มนุษย์เขียนขึ้นจะเปรียบเทียบได้กับโปรแกรมที่ถูกเขียนให้มีเพียงฟังก์ชันหลักเพียงฟังก์ชันเดียวโดยไม่มีการเรียกใช้ฟังก์ชันอื่นๆ วิธีเอดีเอฟซึ่งเสนอโดย Koza (1994) มีแนวทางที่แตกต่างไปจากกระบวนกำหนดการเชิงพันธุกรรมปกติ โดยกำหนดให้ผลเฉลยที่ถูกสร้างขึ้นทั้งหมดในกระบวนการกำหนดการเชิงพันธุกรรมประกอบด้วยโปรแกรมหลักและโปรแกรมย่อย หากเปรียบเทียบกับโปรแกรมคอมพิวเตอร์ที่มนุษย์เขียนขึ้นจะมีลักษณะเป็นโปรแกรมเชิงโครงสร้างซึ่งประกอบไปด้วยฟังก์ชันหลักและฟังก์ชันย่อย ฟังก์ชันหลักสามารถเรียกใช้ฟังก์ชันย่อยได้ และฟังก์ชันย่อยก็สามารถเรียกใช้ฟังก์ชันย่อยอื่นได้เป็นลำดับชั้น

การใช้วิธีเอดีเอฟในกระบวนการกำหนดการเชิงพันธุกรรมจะต้องมีการกำหนดปัจจัยพื้นฐาน 3 ประการคือ

1. จำนวนฟังก์ชันเอดีเอฟในแต่ละผลเฉลย
2. จำนวนอาร์กิวเมนต์ของแต่ละฟังก์ชันเอดีเอฟ
3. ในกรณีที่มีมากกว่า 1 ฟังก์ชันเอดีเอฟต้องมีการกำหนดลำดับชั้นของการอ้างอิงระหว่างฟังก์ชัน

การกำหนดจำนวนฟังก์ชันและอาร์กิวเมนต์เป็นส่วนที่มีความสำคัญมากสำหรับวิธีเอดีเอฟ การเลือกจำนวนฟังก์ชันและจำนวนอาร์กิวเมนต์ที่ไม่เหมาะสมกับปัญหาจะทำให้กระบวนการเรียนรู้มีประสิทธิภาพต่ำลง การเลือกจำนวนฟังก์ชันและอาร์กิวเมนต์ที่เหมาะสมจึงจำเป็นต้องมีการศึกษาปัญหามาเป็นอย่างดี

เนื่องจากผลเฉลยของวิธีเอดีเอฟประกอบไปด้วยหลายฟังก์ชัน ในการสุ่มเลือกผลเฉลย 2 ผลเฉลยเพื่อทำการไขว้เปลี่ยนกันนั้นจึงต้องมีขั้นตอนในการสุ่มเลือกอีกครั้ง เพื่อเลือกว่าการไขว้เปลี่ยนนั้นจะเกิดที่ฟังก์ชันใดของผลเฉลยทั้ง 2 โปรแกรม เนื่องจากโครงสร้างภายใน(ชุดของฟังก์ชันและเทอมินอล) ของแต่ละฟังก์ชันจะมีลักษณะที่แตกต่างกันซึ่งเป็นผลมาจากการกำหนดการอ้างอิงระหว่างฟังก์ชัน จึงไม่สามารถเกิดไขว้เปลี่ยนระหว่างต่างฟังก์ชันซึ่งมีโครงสร้างต่างกันได้

เพื่อแสดงให้เห็นความแตกต่างระหว่างกระบวนกำหนดการเชิงพันธุกรรมปกติและกระบวนกำหนดการเชิงพันธุกรรมที่ใช้วิธีเอดีเอฟ ในที่นี้ขอยกตัวอย่างปัญหาภาวะคู่ (even parity) ซึ่งนำมาจากหนังสือ Genetic Programming II ของ Koza(1994) หน้า 158-194

ปัญหาภาวะคู่มีจุดประสงค์เพื่อหาผลเฉลยที่สามารถตรวจสอบการเป็นหรือไม่เป็นภาวะคู่ของตัวแปรเชิงตรรกซึ่งเป็นอิสระกัน ในกรณีที่มีตัวแปรเชิงตรรกจำนวน 3 ตัว (D0,D1 และ D2) จะทำให้เกิดกรณีที่แตกต่างกันจำนวน 8 กรณี กรณีต่างๆและผลลัพธ์ที่ต้องการของแต่กรณีแสดงเป็นตารางความจริงดังตารางที่ 1.2

ตารางที่ 1.2 ตารางความจริงของปัญหาภาวะคู่ 3 ตัวแปร

กรณีที่	D0	D1	D2	**ผลลัพธ์ที่ต้องการ**
1	T	T	T	F
2	T	T	F	T
3	T	F	T	T
4	T	F	F	F
5	F	T	T	T
6	F	T	F	F
7	F	F	T	F
8	F	F	F	T

สำหรับกระบวนการกำหนดการเชิงพันธุกรรมปกติสำหรับปัญหาภาวะคู่ 3 ตัวแปรมีดังนี้

1. **การสร้างประชากรของผลเฉลยเริ่มต้น** ทำการสร้างประชากรผลเฉลยเริ่มต้นจำนวน 16,000 โปรแกรม ชุดของฟังก์ชันและเทอมินอลที่ใช้ในผลเฉลยคือ

ฟังก์ชัน ฟังก์ชันที่ใช้เป็นฟังก์ชันเชิงตรรกได้แก่ AND, OR, NAND, และ NOR

เทอมินอล ได้แก่ D0, D1 และ D2

2. **การตรวจสอบประสิทธิภาพของผลเฉลย** ทำการวัดค่าความเหมาะของแต่ละผลเฉลยโดยการทดลองกับกรณีทั้ง 8 กรณี จำนวนกรณีที่ได้ผลลัพธ์ถูกต้องคือค่าความเหมาะของผลเฉลยนั้น

3. **การสร้างประชากรของผลเฉลยรุ่นใหม่** ทำการสร้างประชากรรุ่นใหม่โดยการปฏิบัติการเชิงพันธุกรรม(genetic operation) ดังนี้

การสืบพันธุ์ คัดเลือกผลเฉลยรุ่นเดิมที่มีค่าความเหมาะสูงที่สุดจำนวน 10 เปอร์เซนต์ของประชากรมาเป็นประชากรรุ่นใหม่

การไขว้เปลี่ยน สุ่มเลือกผลเฉลยรุ่นเดิมครั้งละ 2 ผลเฉลยเพื่อไขว้เปลี่ยนเป็นประชากรรุ่นใหม่จำนวน 90 เปอร์เซนต์ของประชากร

การกลาย ไม่ใช้การกลายในการทดลองนี้

จากนั้นจึงทำการตรวจสอบประสิทธิภาพและสร้างประชากรรุ่นใหม่ต่อไปจนครบจำนวนรุ่น โดยจำนวนรุ่นที่ทำการทดลองกำหนดไว้ 51 รุ่น จำนวนรอบของการทดลองเท่ากับ 34 รอบ ค่าความเพียรพยายามเชิงคำนวณที่คำนวณได้เท่ากับ 96,000

สำหรับกระบวนการกำหนดการเชิงพันธุกรรมที่ใช้วิธีเอดีเอฟสำหรับปัญหาภาวะคู่ 3 ตัวแปรมีดังนี้

1. **การกำหนดจำนวนฟังก์ชันเอดีเอฟ จำนวนอาร์กิวเมนต์ และการอ้างอิงระหว่างฟังก์ชัน** กำหนดให้โครงสร้างผลเฉลยมี 2 ฟังก์ชันเอดีเอฟ (ADF0 และ ADF1) แต่ละฟังก์ชันมี 2 อาร์กิวเมนต์ การอ้างอิงระหว่างฟังก์ชันเอดีเอฟเป็นแบบลำดับขั้น โดยฟังก์ชันหลักสามารถเรียกใช้ฟังก์ชัน ADF0 และ ADF1 และฟังก์ชัน ADF1 สามารถเรียกใช้ฟังก์ชัน ADF0 ได้

2. **การสร้างประชากรของผลเฉลยเริ่มต้น** ทำการสร้างประชากรผลเฉลยเริ่มต้นจำนวน 16,000 โปรแกรม ชุดของฟังก์ชันและเทอมินอลที่ใช้ในแต่ละฟังก์ชันของผลเฉลยจะแตกต่างซึ่งเป็นผลมาจากการอ้างอิงเชิงลำดับขั้นดังนี้

ฟังก์ชันหลัก ฟังก์ชัน = AND, OR, NAND, NOR, ADF0 และ ADF1

เทอมินอล = D0, D1 และ D2

ฟังก์ชัน ADF0 ฟังก์ชัน = AND, OR, NAND, และ NOR

เทอมินอล = D0, D1 และ D2

ฟังก์ชัน ADF1 ฟังก์ชัน = AND, OR, NAND, NOR และ ADF0

เทอมินอล = D0, D1 และ D2

3. **การตรวจสอบประสิทธิภาพของผลเฉลย** ทำการวัดค่าความเหมาะของแต่ละผลเฉลยโดยการทดลองกับกรณีทั้ง 8 จำนวนกรณีที่ได้ผลลัพธ์ถูกต้องคือค่าความเหมาะของผลเฉลยนั้น

4. **การสร้างประชากรของผลเฉลยรุ่นใหม่** ทำการสร้างประชากรรุ่นใหม่โดยการปฏิบัติการเชิงพันธุกรรมดังนี้

การสืบพันธุ์ คัดเลือกผลเฉลยรุ่นเดิมที่มีค่าความเหมาะสูงที่สุดจำนวน 10 เปอร์เซนต์ของประชากรมาเป็นประชากรรุ่นใหม่

การไขว้เปลี่ยน สุ่มเลือกผลเฉลยรุ่นเดิมครั้งละ 2 ผลเฉลยเพื่อไขว้เปลี่ยน โดยจะมีการสุ่มเลือกว่าการไขว้เปลี่ยนจะเกิดที่ฟังก์ชันหลัก ฟังก์ชัน ADF0 หรือฟังก์ชัน ADF1 ของผลเฉลยทั้ง 2 ผลเฉลยข้างต้น ประชากรผลเฉลยรุ่นใหม่ที่เกิดจากการไขว้เปลี่ยนนี้มีจำนวน 90 เปอร์เซนต์ของประชากรทั้งหมด

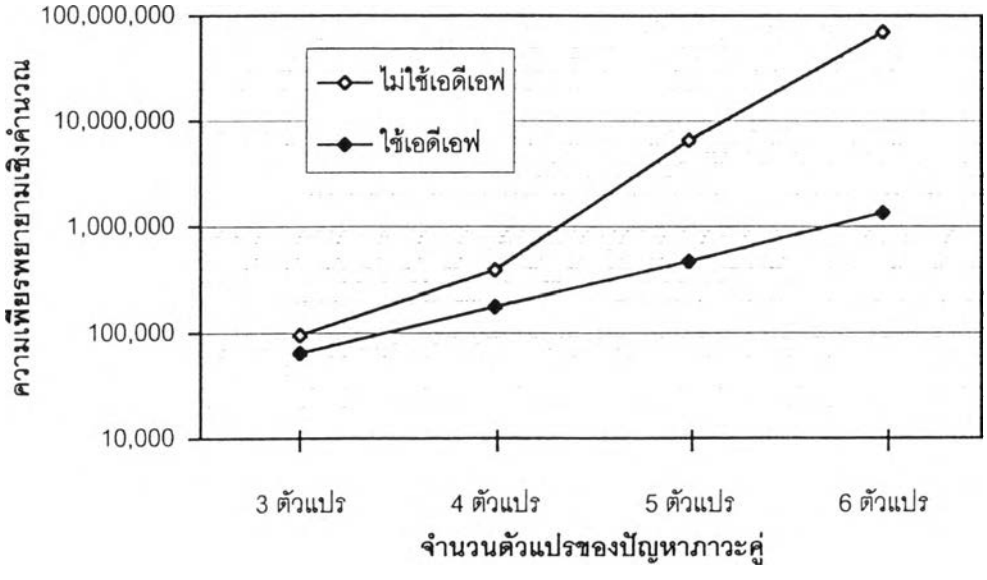
การกลาย ไม่ใช้การกลายในการทดลองนี้

จากนั้นจึงทำการตรวจสอบประสิทธิภาพและสร้างประชากรรุ่นใหม่ต่อไปจนครบจำนวนรุ่น โดยจำนวนรุ่นที่ทำการทดลองกำหนดไว้ 51 รุ่น จำนวนรอบของการทดลองเท่ากับ 33 รอบ ค่าความเพียรพยายามเชิงคำนวณที่คำนวณได้เท่ากับ 64,000 ซึ่งลดลงจากเมื่อไม่ได้ใช้วิธีเอดีเอฟ 33.3 เปอร์เซนต์

เมื่อปัญหาภาวะคู่ 3 ตัวแปรถูกเพิ่มความยากขึ้นเป็นปัญหาภาวะคู่ 4 ตัวแปร ปัญหาภาวะคู่ 5 ตัวแปร และปัญหาภาวะคู่ 6 ตัวแปรพบว่าความเพียรพยายามเชิงคำนวณเมื่อมีการใช้วิธีเอดีเอฟเพิ่มขึ้นในอัตราที่น้อยกว่ากำหนดการเชิงพันธุกรรมปกติดังแสดงในตารางที่ 1.3 และรูปที่ 1.5

ตารางที่ 1.3 ความเพียรพยายามเชิงคำนวณของปัญหาภาวะคู่

ความเพียรพยายามเชิงคำนวณ	ปัญหาภาวะคู่ 3 ตัวแปร	ปัญหาภาวะคู่ 4 ตัวแปร	ปัญหาภาวะคู่ 5 ตัวแปร	ปัญหาภาวะคู่ 6 ตัวแปร
เมื่อไม่ใช้เอดีเอฟ	96,000	384,000	6,528,000	70,176,000
เมื่อใช้เอดีเอฟ	64,000	176,000	464,000	1,344,000



รูปที่ 1.5 กราฟแสดงความเพียรพยายามเชิงคำนวณของปัญหาภาวะคู่

การที่วิธีเอดีเอฟสามารถลดทอนความเพียรพยายามเชิงคำนวณได้อย่างมีประสิทธิภาพเนื่องมาจากวิธีเอดีเอฟใช้หลักการของการแยกส่วนของปัญหาในกระบวนการเรียนรู้แบบกำหนดการเชิงพันธุกรรม กล่าวคือปัญหาจะถูกแบ่งออกเป็นส่วนย่อยหลายๆส่วน ซึ่งง่ายต่อการแก้ปัญหา อาจเปรียบเทียบได้กับโปรแกรมเชิงลำดับชั้นซึ่งเขียนโดยมนุษย์ ฟังก์ชันแต่ละฟังก์ชันจะทำหน้าที่แก้ปัญหาย่อยแต่ละปัญหาโดยเฉพาะ เมื่อรวมฟังก์ชันเหล่านี้เป็นโปรแกรมคอมพิวเตอร์ก็จะทำให้สามารถแก้ปัญหาใหญ่ได้ทั้งหมด การเขียนโปรแกรมเชิงลำดับชั้นยังมีข้อดีที่ช่วยลดความซ้ำซ้อนในการเขียนโปรแกรม การใช้วิธีเอดีเอฟในกระบวนการเรียนรู้แบบกำหนดการเชิงพันธุกรรมจึงสามารถค้นหาส่วนที่ซ้ำซ้อนของปัญหาในลักษณะต่างๆได้ โดยสรุปวิธีเอดีเอฟเป็นวิธีที่มีประสิทธิภาพสำหรับการแก้ปัญหาโดยเฉพาะสำหรับปัญหาที่ซับซ้อน และในกรณีที่ปัญหามีความซับซ้อนมากขึ้น วิธีเอดีเอฟจะมีอัตราการ

เพิ่มของความเพียรพยายามเชิงคำนวณน้อยกว่ากระบวนการเรียนรู้แบบกำหนดการเชิงพันธุกรรมแบบปกติ

สำหรับโครงร่างของวิทยานิพนธ์ฉบับนี้ ในบทที่ 2 จะกล่าวถึงลักษณะของปัญหาแขนหุ่นยนต์และรายละเอียดต่างๆของกำหนดการเชิงพันธุกรรมสำหรับปัญหาแขนหุ่นยนต์ที่ใช้ในการทดลอง ในบทที่ 3 จะเสนอขั้นตอนในการทดลองเพื่อหาวิธีลดทอนความเพียรพยายามเชิงคำนวณที่มีประสิทธิภาพ และสรุปผลการทดลองในบทที่ 4