สำนักงานวิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

ทฤษฎีที่เกี่ยวข้อง

เนื้อหาในบทนี้แบ่งออกเป็น 4 ส่วน คือ 1) ทฤษฎีการหาค่าความเหมาะสม (Optimization) 2) ทฤษฎีกรรมวิธีพันธุกรรม (Genetic Algorithm, GA) 3) ทฤษฎีโครงข่ายใยประสาทเทียม (Artificial Neural Network, ANN) และ 4) ค่าสถิติที่ใช้วัดประสิทธิภาพของแบบจำลอง

3.1 การหาค่าความเหมาะสม

การหาค่าความเหมาะสมเป็นกระบวนการ (Process) ของการค้นหาเงื่อนไข (Conditions) ซึ่งให้ค่ามากที่สุดหรือน้อยที่สุดของฟังก์ชัน โดยปัญหามีข้อจำกัดในการหาค่าความเหมาะสมสามารถเขียนแทนฟังก์ชั่นวัตถุประสงค์ (Objective Function) ได้ด้วยสมการที่ 3-1 (Rao, 1985)

$$\text{หาค่าต่ำสุด } X = \begin{Bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_n \end{Bmatrix} \qquad (3\text{-}1)$$

ภายใต้ข้อจำกัด (Constraints)

$g_j(X) = 0, j = 1, 2 \ldots, m$

และ $l_j(X) = 0, j = 1, 2 \ldots, p$

โดยที่ X = ขนาดเวกเตอร์

$f(X)$ = ฟังก์ชั่นวัตถุประสงค์

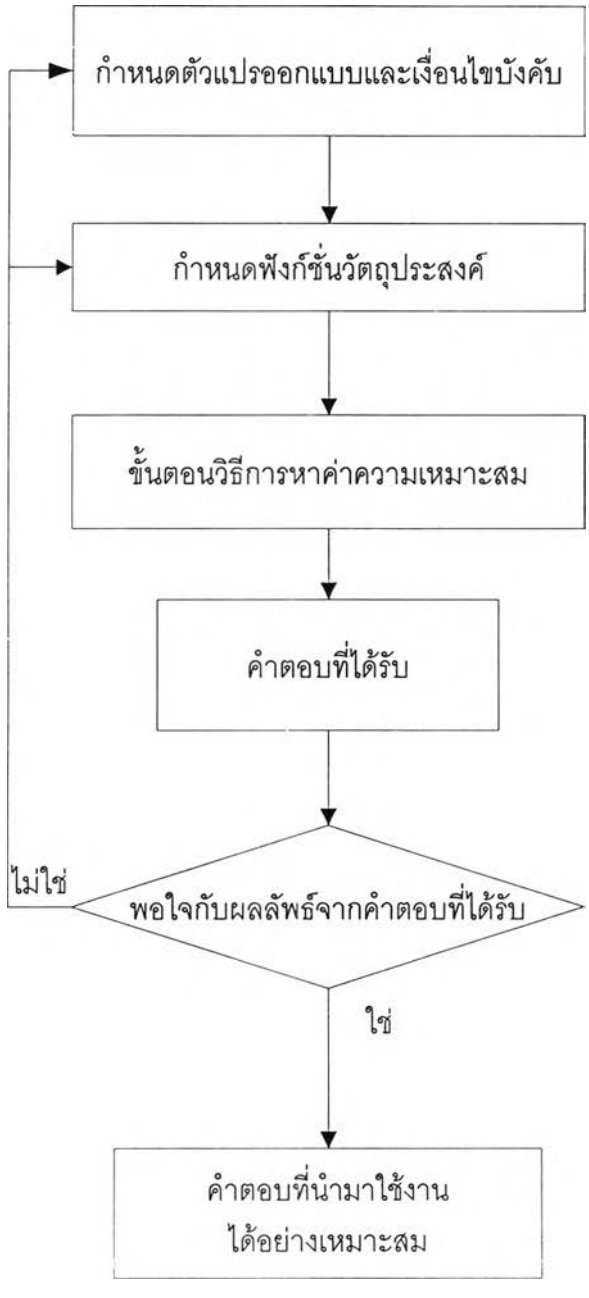
$g_j(X)$ = เงื่อนไขไม่เท่าเทียม (Inequality Constraints)

$l_j(X)$ = เงื่อนไขเท่าเทียม (Equality Constraints)

n = จำนวนตัวแปร

m, p = จำนวนฟังก์ชั่น $g(X)$ และ $l(X)$

การแก้ไขปัญหาในการออกแบบเชิงวิศวกรรมหรือในด้านอื่นๆ โดยใช้วิธีการหาค่าความเหมาะสมที่สุด มักเป็นการหาค่าต่ำสุดหรือจุดสูงสุดของค่าฟังก์ชั่นวัตถุประสงค์ โดยที่มีเงื่อนไขบังคับที่เนื่องมาจากข้อจำกัดต่างๆ ของการออกแบบนั้น ซึ่งเรียกว่า การหาค่าความเหมาะสมแบบมีเงื่อนไขบังคับ (Constrained Optimization) และแผนผังลำดับขั้นของการหาค่าความเหมาะสมแสดงดังแสดงในรูปที่ 3-1



รูปที่ 3-1 แผนผังลำดับขั้นของการหาค่าความเหมาะสม

ที่มา : ธนัญชัย ลีภักดิ์ปรีดา (2543)

3.1.1 วิธีการหาค่าความเหมาะสม

ในปัจจุบันวิธีการหาค่าความเหมาะสมสำหรับปัญหาต่างๆ อาจแบ่งออกได้เป็น 3 แนวทาง (Goldberg, 1989) ดังรูปที่ 3-2 โดยมีรายละเอียดดังนี้

(1) วิธีการคำนวณ (Calculus-Based Methods) เป็นวิธีที่อาศัยการคำนวณโดยตรงจึงต้องอาศัยการศึกษาและเรียนรู้เป็นอย่างมาก วิธีการคำนวณนี้สามารถแบ่งออกได้ 2 วิธีการ คือ

(ก) วิธีโดยอ้อม (Indirect Method) เป็นวิธีการหาค่าสูงสุดหรือต่ำสุดเฉพาะจุด (Local Extrema) โดยทั่วไปเป็นวิธีการหาคำตอบของสมการไม่เชิงเส้น (Nonlinear) ซึ่งทำได้โดยการกำหนดให้ค่าความลาดชัน (Gradient) ของฟังก์ชั่นวัตถุประสงค์ (Objective Function) เท่ากับศูนย์

(ข) วิธีโดยตรง (Direct Method) เป็นวิธีการหาค่าความเหมาะสมเฉพาะจุด (Local Optima) ซึ่งทำได้โดยใช้วิธีกำหนดให้ค่าอนุพันธ์ของสมการแล้วเลื่อนตำแหน่งการคำนวณไปตามทิศทางที่มีความสัมพันธ์กับความลาดชันเฉพาะที่ (Local Gradient) ซึ่งคล้ายกับการไต่เขา (Hill Climbing) ซึ่งในการหาค่าที่ดีที่สุดเฉพาะจุด ทำได้โดยกำหนดให้ไต่ฟังก์ชั่นไปในทิศทางที่มีความชันมากที่สุดที่เป็นไปได้ (Steepest Permissible)

วิธีการคำนวณได้ถูกพัฒนาและแก้ไขมาแล้ว แต่ยังคงมีข้อเหตุผลอย่างง่ายที่แสดงให้เห็นว่า วิธีการคำนวณยังขาดความเชื่อมั่น (Robustness) คือ วิธีการนี้เป็นการหาค่าความเหมาะสมเฉพาะจุด การคำนวณเป็นการหาคำตอบจากค่าตัวแปรอิสระเพียงช่วงใดช่วงหนึ่งที่สนใจเท่านั้นไม่ใช่จากค่าตัวแปรอิสระทั้งหมด ดังนั้น คำตอบที่ได้อาจเป็นคำตอบที่ดีที่สุดสำหรับช่วงค่าตัวแปรอิสระนั้น ๆ และคำตอบที่ได้นั้นอาจไม่ใช่คำตอบที่แท้จริง หากฟังก์ชั่นมีจุดสูงสุด-ต่ำสุดเพียงยอดเดียว (Single-Peak) การคำนวณจะทำได้โดยง่าย แต่ถ้าฟังก์ชั่นมีจุดสูงสุด-ต่ำสุดหลายยอด (Multiple-Peak) นั้นอาจทำให้การคำนวณเกิดความผิดพลาดขึ้นได้ และเนื่องจากวิธีการคำนวณนี้เป็นการหาความเหมาะสมโดยอาศัยการหาค่าอนุพันธ์ของฟังก์ชั่น ซึ่งพบว่าบางครั้งการหาค่าอนุพันธ์ของฟังก์ชั่นทำได้ยากและใช้เวลามากกว่าการหาคำตอบด้วยวิธีประมาณเชิงตัวเลข (Numerical Approximation) ซึ่งคำตอบที่ได้จากวิธีประมาณเชิงตัวเลขนั้นสามารถนำไปใช้งานได้เช่นกัน

(2) วิธีแจงนับ (Enumerative Schemes)

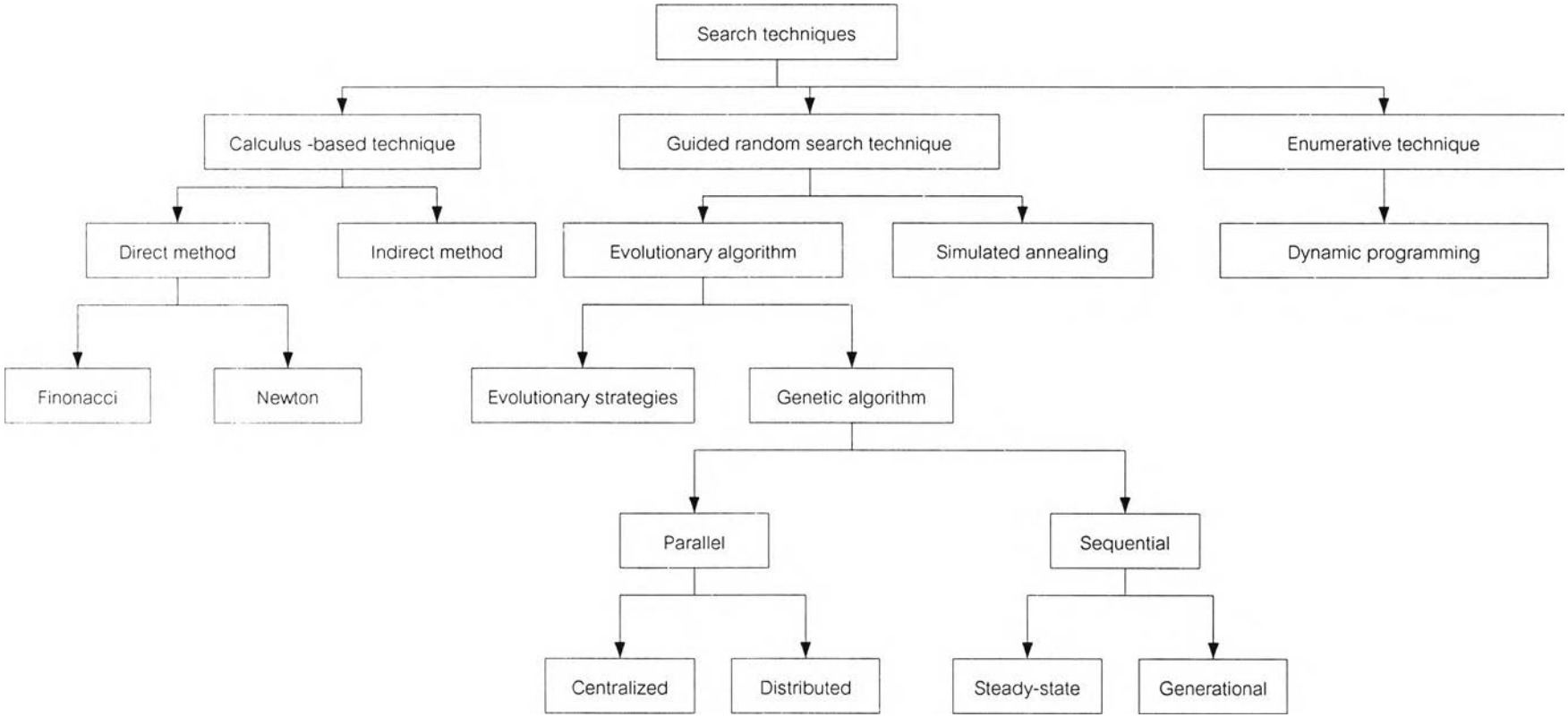
เป็นวิธีการที่ค้นหาคำตอบของฟังก์ชั่นวัตถุประสงค์ จากจุดทุกจุดในอาณาเขตที่เป็นไปได้ทั้งหมด (Search Space) วิธีนี้เหมาะสมกับปัญหาที่มีจำนวนคำตอบที่เป็นไปได้จำนวนน้อย หรือ จำนวนคำตอบที่มีจำนวนมาก แต่แยกออกจากกัน (Discretised Infinite Search Space) ทำให้วิธีการนี้ไม่เหมาะสมกับการแก้ปัญหาที่มีความซับซ้อนหรือมีจำนวนคำตอบที่เป็นไปได้จำนวน มาก ๆ และเป็นวิธีการที่ต้องอาศัยความสัมพันธ์เป็นขั้นตอน ซึ่งมากเกินความจำเป็นอีกทั้งต้องมีความรู้พิเศษบางประการในการแก้ปัญหาด้วย เช่น โปรแกรมเชิงพลวัต (Dynamic Programming) เป็นต้น

(3) วิธีการค้นหาค่าแบบสุ่ม (Random Search Algorithm)

วิธีการค้นหาค่าแบบสุ่มนี้ได้รับความนิยมเพิ่มมากขึ้นเรื่อย ๆ เนื่องจากนักวิจัยหลายท่านเข้าใจถึงจุดอ่อนของวิธีการคำนวณ และวิธีการแจงนับ แต่อย่างไรก็ตามสำหรับการคำนวณที่ใช้จำนวนรอบมาก ๆ ของวิธีการค้นหาค่าแบบสุ่มนั้นสามารถคาดได้ว่าคำตอบที่ได้เป็นค่าที่ไม่ดีไปกว่าคำตอบที่ได้จากการแจงนับ วิธีการค้นหาค่าแบบสุ่ม ควรแบ่งแยกออกจากเทคนิคการสุ่ม (Randomized Techniques) เพราะเทคนิคการสุ่มนั้นไม่จำเป็นต้องมีทิศทางของการค้นหาคำตอบ แต่วิธีการค้นหาค่าแบบสุ่มนั้นมีเครื่องมือที่นำไปสู่คำตอบ ตัวอย่างวิธีการค้นหาแบบสุ่ม เช่น GA ที่ใช้วิธีการใส่รหัสข้อมูลพันธุ์ (Coding) การคัดเลือกพันธุ์ (Selection) ผสมข้ามพันธุ์ (Cross over) และ ปรับปรุงพันธุ์ (Mutation) และอาศัยการกำหนดค่าพารามิเตอร์ต่าง ๆ ที่เหมาะสมอันเครื่องมือทำให้เกิดการลู่เข้าหาคำตอบ เป็นต้น

3.2 ทฤษฎีกรรมวิธีพันธุกรรม (Genetic Algorithm, GA)

ประมาณทศวรรษที่ 1960 ได้มีความสนใจเพิ่มมากขึ้นในการพัฒนาความสามารถของขั้นตอนทางคอมพิวเตอร์ (Algorithm) โดยใช้วิธีการเลียนแบบสิ่งมีชีวิตในการวิวัฒนาการให้ได้พันธุ์ที่ดีที่สุดเพื่อนำไปใช้ในการหาค่าความเหมาะสมที่สุดของปัญหาที่มีความซับซ้อนและยุ่งยาก ซึ่งวิธีที่รู้จักกันโดยทั่วไป และจัดอยู่ในประเภทนี้ได้แก่ Genetic Algorithm พัฒนาขึ้นโดย Holland (1975), Evolution Strategies พัฒนาขึ้นโดย Rechenberg (1973) และ Schwefel (1995), Evolution Programming พัฒนาขึ้นโดย Fogel et al (1966) และ Genetic Programming พัฒนาขึ้นโดย Koza (1992) เป็นต้น (Mitsuo และ Runwei, 1997)



รูปที่ 3-2 วิธีการหาค่าความเหมาะสม

ที่มา : http://www.web.umr.edu/~ercal/387/slides/GATutorial.ppt

3.2.1 พันธุศาสตร์กับกรรมวิธีพันธุกรรม

เมนเดล บิดาแห่งวิชาพันธุศาสตร์ ค้นพบว่าลักษณะต่างๆของสิ่งมีชีวิต เช่น ลักษณะผิวของเมล็ดพืช สีของเมล็ดพืช ฯลฯ ที่ถูกถ่ายทอดไปยังลูกหลานนั้นถูกควบคุมโดยหน่วยควบคุมลักษณะที่เรียกว่า ยีน (Gene) และลักษณะย่อยของยีนเรียกว่า อัลลีล (Allele) เช่น ยีนควบคุมลักษณะผิวของเมล็ดจะมีอัลลีลเป็นผิวขรุขระและผิวเรียบ เป็นต้น ซึ่งแต่ละยีนจะเรียงตัวเป็นสายที่เรียกว่า โครโมโซม (Chromosome)

ชาร์ล ดาร์วิน (1859) ได้เสนอความคิดการเกิดชนิดของสิ่งมีชีวิต (The Original of Species) โดยเสนอหลักการของวิวัฒนาการที่ผ่านกระบวนการคัดเลือกตามธรรมชาติ แม้ในตอนแรกทฤษฎีจะเป็นที่โต้แย้งกันมาก ต่อมาก็ได้เป็นที่ยอมรับในหมู่นักวิทยาศาสตร์ (Winston, 1992)

- สิ่งมีชีวิตแต่ละชนิดมีแนวโน้มที่จะถ่ายทอดลักษณะของมันไปสู่ลูกหลานของมัน

- ธรรมชาติทำให้สิ่งมีชีวิตมีลักษณะที่ต่างกัน

- สิ่งมีชีวิตที่มีความเหมาะสม ซึ่งมีลักษณะที่เหมาะสมที่สุดมีแนวโน้มที่จะมีลูกหลานมากกว่าสิ่งมีชีวิตที่มีลักษณะไม่เหมาะสม ซึ่งจะทำให้ประชากรอยู่รอดต่อไป

- เมื่อระยะเวลาผ่านไปยาวนานจะเกิดการกลายพันธุ์ขึ้นและเกิดชนิดใหม่ที่มีลักษณะเหมาะสมกับระบบนิเวศน์นั้น

Holland และคณะผู้ร่วมงานที่มหาวิทยาลัยมิชิแกน ประเทศสหรัฐอเมริกา ได้วิจัยและพัฒนากรรมวิธีพันธุกรรม (Genetic Algorithm, GA) ขึ้นจนสำเร็จและเผยแพร่เมื่อปี ค.ศ. 1975 โดยมีวัตถุประสงค์ของการวิจัย 2 ประการ คือ (Goldberg, 1989)

(1) เพื่อสรุปและดัดแปลงการใช้กระบวนการทางธรรมชาติให้ถูกต้องมากที่สุด

(2) ออกแบบและสร้างซอฟต์แวร์ที่รักษากลไกที่สำคัญของธรรมชาติ

ซึ่งการวิจัยนี้นำไปสู่การค้นพบที่สำคัญของวิทยาศาสตร์ธรรมชาติ (Natural Science) และวิทยาศาสตร์ประดิษฐ์ (Artificial System Science)

3.2.2 กรรมวิธีพันธุกรรม

กรรมวิธีพันธุกรรม (GA) เป็นวิธีการค้นหาคำตอบโดยมีพื้นฐานแนวความคิดมาจากกระบวนการคัดเลือกโดยธรรมชาติของ ชาร์ล ดาร์วิน (Charles Darwin) และกระบวนการคัดเลือกทางพันธุศาสตร์ (Natural Genetics Selection) เป็นการใช้วิธีการเชิงความเป็นไปได้ในการอยู่รอดของสิ่งที่เหมาะสมที่สุด โดยอาศัยกระบวนการ คัดเลือกพันธุ์ (Selection), ผสมข้ามพันธุ์ (Crossover) และ ปรับปรุงพันธุ์ (Mutation) ซึ่งประสิทธิภาพของกระบวนการเหล่านี้ขึ้นอยู่กับวิธีใช้และการกำหนดค่าพารามิเตอร์ต่างๆ ที่ทำให้การคำนวณมีลู่เข้าหาคำตอบมากขึ้น

เมื่อนำ GA ไปเปรียบเทียบกับวิธีการหาความเหมาะสมอื่น ๆ เช่น โปรแกรมเชิงเส้น (Linear Programming), โปรแกรมเชิงพลวัต (Dynamic Programming) พบว่า มีความแตกต่างกัน ซึ่งทำให้ GA ได้เปรียบกว่าวิธีการอื่น ๆ ดังนี้ (Goldberg, 1989)

(1) GA ดำเนินการด้วยการแปลงรูปแบบของตัวพารามิเตอร์ หรือการกำหนดรหัสพันธุ์ไม่ได้ใช้ตัวพารามิเตอร์โดยตรง ซึ่งมีประโยชน์ในกรณีที่มีตัวแปรมาก หรือตัวแปรมีลักษณะที่ไม่สามารถหาค่าความเหมาะสมได้พร้อมกัน

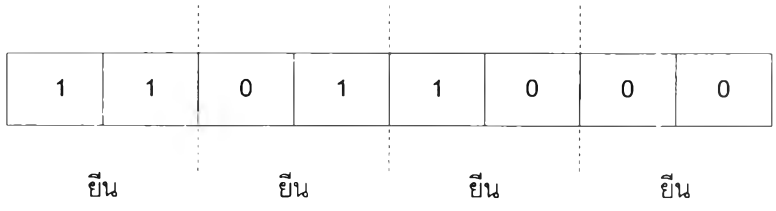
(2) GA ดำเนินการหาคำตอบจากประชากรของทางเลือกทั้งหมดที่เป็นไปได้ ไม่ได้ค้นหาคำตอบจากจุดใดจุดหนึ่ง ดังนั้นโอกาสที่พบคำตอบที่เป็นค่าที่เหมาะสมสูงสุดที่แท้จริงจึงมีมากขึ้น

(3) GA ใช้ค่าฟังก์ชั่นวัตถุประสงค์ (Objective Function) เป็นเพียงแนวทางและตัวชี้นำเท่านั้น ไม่ได้หาคำตอบจากตัวฟังก์ชั่นวัตถุประสงค์โดยตรง จึงเหมาะสมกับฟังก์ชั่นที่ยุ่งยากซับซ้อน

(4) GA ใช้วิธีการเชิงความน่าจะเป็น (Probabilistic) ไม่ใช่เชิงกำหนดคงตัว (Deterministic)

การหาค่าความเหมาะสมด้วย GA ตัวแปรต่างๆจะถูกแปลง (Coding) ให้อยู่ในรูปของแถวของตัวเลข (String) หรือ โครโมโซม (Chromosome) โดยในแต่ละโครโมโซมประกอบด้วยบล็อกหรือหน่วยถ่ายพันธุ์ ที่เรียกว่า ยีน (Gene) ที่เป็นอักขระ (Character) หรือ บิต (Bit) ดังแสดงในรูปที่ 3-3 ยีนแต่ละยีนนั้นเป็นตัวแปรตัดสินใจ (Decision Variables) ที่ทำให้ฟังก์ชั่นวัตถุประสงค์ มีค่าสูงสุดหรือต่ำสุดได้

หอสมุดกลาง สำนักงานวิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3-3 โครโมโซม1 ตัวประกอบด้วยยีน 4 ตัว

ในระยะแรกของการพัฒนา GA นั้น ยีนในโครโมโซมถูกแทนค่าด้วย 0 หรือ 1 หรือเรียกว่า Binary Bits (Goldberg และ Kao, 1987; Wang, 1991) ซึ่งเมื่อถอดรหัส (Decode) ออกมาจะเป็นค่าของตัวแปรตัดสินใจ โดยยีนในรูปแบบดังกล่าวอาจแปลงค่าหรือใส่รหัสได้ในหลาย ๆ แบบ เช่น จำนวนจริง (Real Number), จำนวนเต็ม (Integer Number) หรือ เซท (Set) เป็นต้น ขึ้นอยู่กับผู้ใช้กำหนดให้เป็นเช่นไร เพื่อให้มีความเหมาะสมกับปัญหาแต่ละปัญหา เมื่อแทนค่าของยีนทุกตัวใน 1 โครโมโซม ลงในฟังก์ชั่นวัตถุประสงค์ โครโมโซมแต่ละตัวจะแสดงถึงผลลัพธ์ที่ได้ (Fitness) โดยทั้งนี้ค่าที่ได้นั้นอาจเป็นเพียงค่าที่เป็นไปได้ หรือเป็นค่าที่ใกล้เคียงกับค่าที่เหมาะสมที่สุด จากนั้นโครโมโซมเหล่านี้จะค่อย ๆ มีการพัฒนาวิวัฒนาการมากขึ้นตามลำดับ โดยอาศัยวิธีการที่เลียนแบบกระบวนการทางธรรมชาติ คือ การคัดเลือกพันธุ์ (Selection), ผสมข้ามพันธุ์ (Crossover) และการปรับปรุงพันธุ์ (Mutation) ซึ่งการวิวัฒนาการของประชากรในรุ่นต่อ ๆ มา (Generation) นี้นำไปสู่การหาค่าความเหมาะสมที่สุดได้นั้นเอง ขั้นตอนการทำงานของ GA แสดงในรูปที่ 3-4 และอธิบายได้ดังนี้

(1) [เริ่มต้น] สร้างประชากรที่ประกอบด้วยโครโมโซมหลายๆตัว ด้วยการสุ่ม (สร้างคำตอบที่เหมาะสมกับปัญหา)

(2) [ประเมินค่า] ประเมินค่าฟังก์ชั่นวัตถุประสงค์ของแต่ละโครโมโซม

(3) [ตรวจสอบ] ถ้าให้ผลที่พอใจหรือครบจำนวนรอบคำนวณให้หยุดการคำนวณ

(4) [สร้างประชากรใหม่] สร้างประชากรชุดใหม่ด้วยขั้นตอนข้างล่างนี้จนกระทั่งได้ประชากรชุดใหม่ครบ

(4.1) [คัดเลือกพันธุ์] คัดเลือกโครโมโซมจากประชากรชุดเดิมโดยพิจารณาจากความเหมาะสม

(4.2) [ผสมข้ามพันธุ์] ด้วยความน่าจะเป็นในการผสมข้ามพันธุ์ทำให้เกิดโครโมโซมใหม่จากโครโมโซมพ่อและแม่

(4.3) [ปรับปรุงพันธุ์] ด้วยความน่าจะเป็นในการปรับปรุงพันธุ์ทำให้เกิดการเปลี่ยนแปลงยีนในโครโมโซมจึงได้โครโมโซมชุดใหม่

(4.4) [แทนที่] แทนที่ประชากรชุดเดิมด้วยประชากรชุดใหม่

(5) [วนซ้ำ] ไปที่ขั้นที่ 2

3.2.3 ขั้นตอนการดำเนินการของ GA (Operator of GA)

ขั้นตอนการดำเนินการของ GA ประกอบด้วย 4 ขั้นตอน ได้แก่ การเข้ารหัสพันธุ์ (Coding) การคัดเลือกพันธุ์ (Selection) การผสมข้ามพันธุ์ (Crossover) และ การปรับปรุงพันธุ์ (Mutation)

(1) การเข้ารหัสพันธุ์ (Coding)

ขั้นตอนแรกของ GA คือ การแปลงค่าตัวแปรให้อยู่ในรูปของสตริงหรือโครโมโซมที่มีความยาวแน่นนอน ซึ่งมีอยู่หลายวิธีซึ่งสามารถอธิบายได้ดังนี้ (Obitko, 1998)

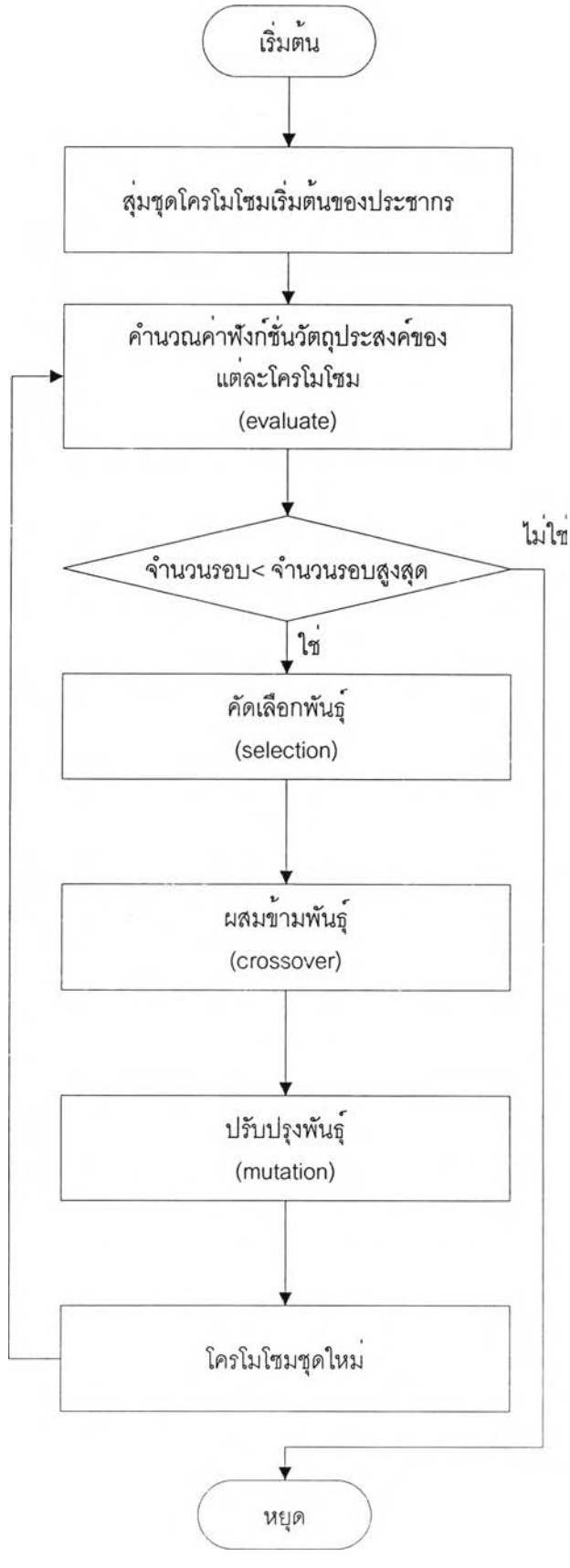
(ก) Binary String Encoding

คำตอบของปัญหาที่ต้องการหาค่าความเหมาะสมนั้นถูกแทนด้วยแถวของ Binary String ซึ่งในแต่ละแถวเรียกว่า โครโมโซม ในโครโมโซม ประกอบด้วย ยีน โดยในแต่ละยีนนั้นเป็นค่าตัวแปรตัดสินใจ (Decision Variable) ในฟังก์ชั่นวัตถุประสงค์ (Objective Function) ที่ทำให้มีค่าสูงสุดหรือต่ำสุด จากตัวอย่าง 1 โครโมโซม ประกอบด้วย 5 ยีน ใน 1 ยีนประกอบด้วย Binary 3 ตัวหรือเรียกว่า 3 อัลลีล (Allele) ซึ่งเมื่อถอดรหัสออกมาเป็นค่าของตัวแปรตัดสินใจแล้วอาจได้ค่าที่เป็น จำนวนจริง, จำนวนเต็ม หรือ เซท ได้แล้วแต่ผู้ใช้ว่า กำหนดให้เป็นเช่นไร ตามความเหมาะสมของแต่ละปัญหาเช่น

โครโมโซม A ประกอบด้วย 110 110 010 011 011

โครโมโซม B ประกอบด้วย 110 111 100 001 111

ข้อดีของ Binary Encoding คือสามารถแปรค่าออกมาเป็นตัวเลขได้หลากหลายแล้วแต่การกำหนดจำนวนหลักของอัลลีล แต่ในทางกลับกันบ่อยครั้งที่วิธีการนี้ไม่เหมาะสมกับปัญหาบางชนิดซึ่งทำให้ในบางครั้งต้องแก้ไขหลังจากที่ Crossover หรือ Mutation แล้ว



รูปที่ 3-4 ขั้นตอนการใช้ GA ในการแก้ปัญหาการหาค่าความเหมาะสม

(ข) Permutation Encoding

Permutation Encoding สามารถใช้กับปัญหาที่มีลักษณะเป็นการเรียงลำดับ (Ordering Problems) เช่น Traveling Salesman Problem และ ปัญหาการจัดสายผลิต เป็นต้น Permutation Encoding เป็นการแทนทุก ๆ โครโมโซมด้วยสตริงที่เป็นตัวเลข ซึ่งตัวเลขเหล่านี้แสดงถึงลำดับขั้นการทำงาน เช่น

โครโมโซม A ประกอบด้วย 1 5 3 2 6 4 7 9 8

โครโมโซม B ประกอบด้วย 8 5 6 7 2 3 1 4 9

แต่ปัญหาลักษณะนี้การกำหนด Crossover และ Mutation ต้องระมัดระวัง เพื่อให้เกิดการเรียงลำดับที่ถูกต้องและไม่เกิดการเรียงลำดับซ้ำซ้อนกันในโครโมโซม

(ค) Value Encoding

เป็นการใส่รหัสค่าโดยตรงจะใช้เมื่อการใช้วิธีการใส่รหัสแบบอื่นๆ เช่น Binary Encoding เป็นต้น ทำให้การแก้ปัญหามีความยุ่งยากซับซ้อนมากขึ้นไปอีก ในการใสรหัสโดยตรง ทุกๆยีนในโครโมโซมแสดงถึง ลำดับของค่าที่มีได้หลากหลายรูปแบบ เช่น

โครโมโซม A ประกอบด้วย 1.23 5.32 0.45 2.32 2.45

โครโมโซม B ประกอบด้วย (หลัง), (หลัง), (ขวา), (ซ้าย), (หน้า)

วิธี Value Encoding มีความเหมาะสมกับปัญหาพิเศษ และเมื่อใช้วิธีการนี้จำเป็นต้องค้นหาและพัฒนา ผสมข้ามพันธุ์ และ ปรับปรุงพันธุ์ ที่เหมาะสมกับปัญหานั้น ๆ และมีความเหมาะสมในการหาค่าถ่วงน้ำหนักของโครงข่าย ANN เนื่องจากการใช้วิธี Binary Coding จะมีขั้นตอนการคำนวณที่มากกว่าและมีตัวแปรที่มากกว่า (Obitko, 1998; Sexton และ Gupta, 2000; Jain, 2005)

(2) การคัดเลือกพันธุ์ (Selection)

โครโมโซมที่มีค่าเหมาะสมที่ดีจะถูกคัดเลือกออกมาจากประชากรทั้งหมดเพื่อนำมาเป็นพ่อและแม่ (Parents) ในการ Crossover วิธีการคัดเลือกนั้นมีอยู่หลายวิธีเพื่อให้ได้โครโมโซมที่ดีที่สุด เช่น Roulette Wheel Selection, Rank Selection, Tournament Selection, Steady State Selection และ Elitism เป็นต้น

(ก) Roulette Wheel Selection

วงล้อรูเล็ต คือ วงกลมที่มีพื้นที่ขนาด 1 หน่วยซึ่งพื้นที่ถูกแบ่งออกเป็นส่วนๆ ตามจำนวนประชากร (Population Size) พื้นที่แต่ละส่วนในวงล้อจะขนาดเท่ากับความน่าจะเป็นในการถูกเลือกของโครโมโซมแต่ละตัว ดังสมการที่ 3-2 คือ โครโมโซมที่มีความเหมาะสมมากจะมีสัดส่วนพื้นที่ในวงล้อมาก ดังรูปที่ 3-5 แสดงว่าในการคัดเลือกหรือการหมุนวงล้อครั้งหนึ่งโอกาสที่วงล้อหยุดตรงกับพื้นที่ของโครโมโซมที่มีความเหมาะสมที่สุดมากทำให้มีการถูกเลือกซ้ำได้มาก

$$p(i) = \frac{f(i)}{\sum f(i)} \qquad (3\text{-}2)$$

โดย $p(i)$ คือ ความน่าจะเป็นในการถูกเลือกของโครโมโซมตัวที่ i

$f(i)$ คือ ค่าความเหมาะสมของโครโมโซมตัวที่ i

(ข) Rank Selection

การคัดเลือกแบบ Roulette Wheel Selection อาจเกิดปัญหาได้เมื่อมีโครโมโซมที่มีความเหมาะสมมาก ดังรูปที่ 3-6 (ก) ซึ่งจะทำให้โครโมโซมอื่นๆ มีโอกาสถูกเลือกน้อย หรือ กล่าวได้ว่า จะได้ถูกโครโมโซมชุดเดียวกันมาจับคู่กันเองมากครั้งเกินไปและคำตอบจะลู่เข้าเร็วเกินไป (Obitko, 1998; Hsu, 1999; Eiben และ Smith, 2003)

วิธี Rank Selection ทำโดยการจัดลำดับค่าความเหมาะสมของทุกโครโมโซม จากค่าที่แย่ที่สุดไปยังค่าที่ดีที่สุด แล้วกำหนดให้ค่าที่ดีที่สุดมีค่าของการจัดลำดับเท่ากับ 1, 2 ...N ตามลำดับ ซึ่งค่าที่แย่ที่สุดมีค่าของการจัดลำดับ เท่ากับ N จากนั้นจะประเมินความน่าจะเป็นในการถูกเลือกสมการต่างๆ ดังเช่น สมการที่ 3-3

$$p(i) = \frac{1}{n}\left[\beta - 2(\beta - 1)\frac{i-1}{n-1}\right], 1 \le \beta \le 2 \qquad (3\text{-}3)$$

โดยที่ i คือ ลำดับการ ranking ของโครโมโซม

n คือ จำนวนโครโมโซม

β คือ ตัวแสดงความแตกต่างจากค่าที่ดีที่สุด

ภาพหลังจากการทำ Ranking ดังแสดงในรูปที่ 3-6 (ข) ทำให้โอกาสการถูกเลือกของแต่ละโครโมโซม มีค่ามากน้อยลดหลั่นกันไปตามค่าลำดับความเหมาะสมและจำนวนโครโมโซม แต่วิธีการนี้อาจทำให้ใช้เวลาในการคำนวณนานมากขึ้น เพราะทำให้ โครโมโซมที่มีค่าดีอาจมีค่าไม่แตกต่างจากโครโมโซมอื่น ๆ ทำให้มีโอกาสถูกเลือกได้น้อยลงกว่าที่ควรเป็น (Obitko, 1998; Eiben และ Smith, 2003)

(ค) Tournament Selection

วิธีการนี้ทำได้โดยสุ่มเลือกโครโมโซมขึ้นมา 1 คู่ แล้วนำค่าความเหมาะสมของทั้งคู่มาพิจารณาเปรียบเทียบโครโมโซมใดมีค่าความเหมาะสมที่ดีกว่าจะถูกเลือกไว้ในรุ่นต่อไป ส่วน โครโมโซมใดที่ให้ค่าไม่ดีจะถูกทิ้งไป แต่ถ้าหากค่าความเหมาะสมของโครโมโซมทั้งสองมีค่าเท่ากัน ทำการสุ่มเลือกโครโมโซมคู่ใหม่ขึ้นมาพิจารณา (Goldberg และ Deb, 1989)

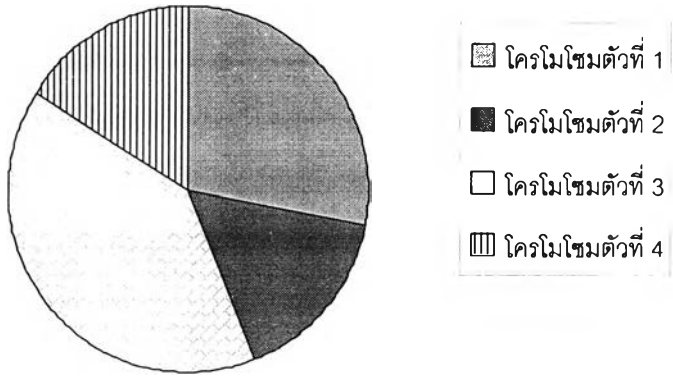
วิธี Tournament Selection เป็นที่นิยมใช้กันมาก เนื่องจากคำนวณได้ง่าย ให้คำตอบที่เร็วกว่าและมีตัวแปรในการคำนวณน้อยเมื่อเทียบกับวิธีอื่นๆ (Ursem, 2003)

(ง) Steady-State Selection

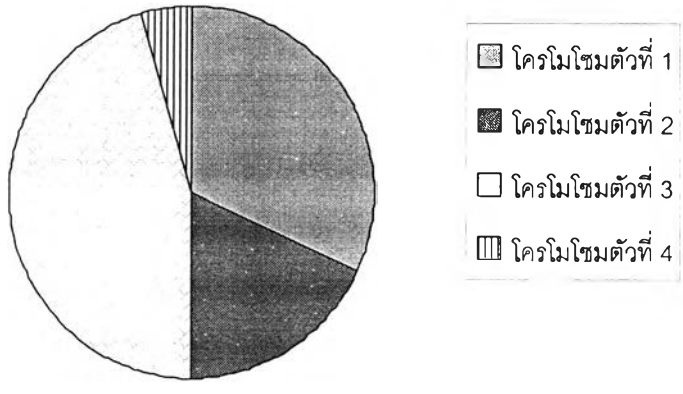
วิธีการนี้ไม่ใช่วิธีการที่ใช้เลือกโครโมโซมพ่อแม่โดยเฉพาะ เนื่องจากวิธีการนี้มีแนวความคิดว่า โครโมโซมส่วนใหญ่ที่มีค่าความเหมาะสมที่ดีควรมีชีวิตอยู่รอดไปในรุ่นถัดไป วิธีการนี้ทำโดยในทุก ๆ รุ่นมีการเลือกโครโมโซมบางตัว (2 - 3 ตัว) ที่มีค่าที่ดีไว้เพื่อสร้างโครโมโซมตัวใหม่ สำหรับโครโมโซมบางตัวที่มีค่าที่ไม่ดีถูกทิ้งโดยใช้โครโมโซมตัวใหม่ ที่มีค่าที่ดีมาแทนที่โครโมโซมเหล่านั้น ประชากรที่เหลืออยู่กลายเป็นประชากรในรุ่นต่อไป

(จ) Elitism

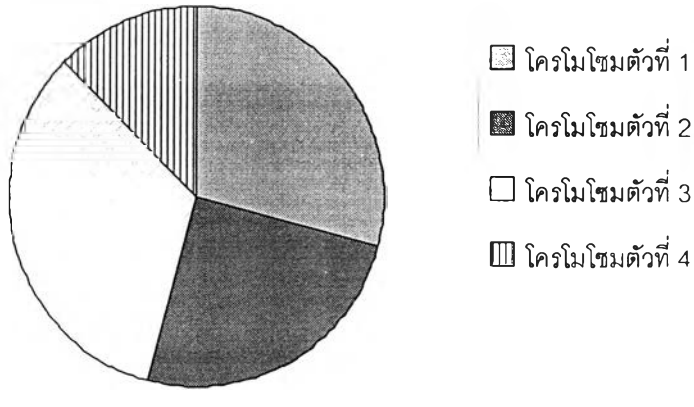
แนวความคิดของวิธี Elitism คือ เมื่อสร้างประชากรรุ่นต่อไปโดยการ Crossover และ Mutation แล้ว ยังมีโอกาสมากที่สูญเสียโครโมโซมที่มีค่าที่ดีที่สุดไป ดังนั้นวิธีการ Elitism ใช้วิธีการคัดลอก (Copy) โครโมโซมที่มีค่าที่ดีไว้ (อาจมีหลายตัว) แล้วนำไปเป็นประชากรในรุ่นต่อไป ส่วน โครโมโซมตัวอื่น ๆ ใช้วิธีอื่นไปตามปกติ วิธี Elitism สามารถทำให้ประสิทธิภาพของ GA เพิ่มมากขึ้น เพราะว่าวิธีการนี้ป้องกันการสูญเสียโครโมโซมที่มีค่าที่ดีไป



รูปที่ 3-5 Roulette Wheel Selection



(ก) ค่าความเหมาะสมก่อนทำ Rank Selection



(ข) ค่าความเหมาะสมหลังทำ Rank Selection

รูปที่ 3-6 Rank Selection

(3) การผสมข้ามพันธุ์ (Crossover)

การผสมข้ามพันธุ์ เป็นขั้นตอนที่สำคัญมากอีกขั้นหนึ่งของ GA หากไม่ทำการผสมข้ามพันธุ์ จะทำให้โครโมโซมต่างๆ คงสภาพเดิมไม่มีการเปลี่ยนแปลงเกิดขึ้น ทำให้การปฏิบัติการของ GA ไม่ได้ผลหรือไม่อาจเรียกเป็น GA ได้ การผสมข้ามพันธุ์ในโครโมโซมเป็นจำนวนเท่าใดขึ้นอยู่กับความน่าจะเป็น คือ Probability of Crossover (Pc) ประเภทของการผสมข้ามพันธุ์ จำแนกได้ 4 แบบ คือ

(ก) Crossover 1 ตำแหน่ง (One-Point Crossover)

(ข) Crossover 2 ตำแหน่ง (Two-Point Crossover)

(ค) Crossover หลายตำแหน่ง (Uniform Crossover)

(ง) Crossover แบบเลขคณิต (Arithmetic Crossover)

Crossover 1 ตำแหน่งนั้น มีการสุ่มจุด 1 จุดในโครโมโซมทำให้แบ่งโครโมโซมออกเป็น 2 ส่วนและมีการแลกเปลี่ยนแถวของยีนระหว่างโครโมโซมพ่อแม่ที่มาจับคู่กัน ณ จุดซึ่งเกิดจากการสุ่ม ส่วน Crossover 2 ตำแหน่งมีการสุ่มสุ่มจุด 2 จุดในโครโมโซม ดังภาพแสดงตำแหน่งการ Crossover ที่สรุปไว้อย่างเข้าใจง่ายดังรูปที่ 3-7 สำหรับวิธีการ Crossover แบบเลขคณิต นั้นจะทำการแลกเปลี่ยนค่าของยีนด้วยสมการทางคณิตศาสตร์ ดังสมการที่ 3-4 (Eiben และ Smith, 2003)

$$Child = (1-\alpha)(Parent1) + \alpha(Parent2) \qquad (3\text{-}4)$$

โดย α คือ ค่าตัวแปรจากการสุ่มมีค่าระหว่าง 0 ถึง 1

ในการศึกษาครั้งนี้ได้เลือกใช้ค่า α ใน 2 รูปแบบและได้กำหนดชื่อเรียกที่แตกต่างกัน คือ ค่า α เท่ากับ 0.5 เรียกชื่อ Average Crossover และ ค่า α ที่ได้จากการสุ่มแต่ละครั้งระหว่าง 0 ถึง 1 เรียกชื่อว่า Heuristic Crossover

(4) การปรับปรุงพันธุ์ (Mutation)

เป็นขั้นตอนที่ยีน ถูกปรับเปลี่ยนดัดแปลงให้ผิดแผกไปจากโครโมโซมพ่อ แม่ โดยสิ้นเชิงภายใต้การควบคุมของ Probability of Mutation (Pm) ในกรณีที่ยีนแสดงค่าเป็นไบนารี่หรือเลขฐานสอง การดัดแปลงยีน คือ การเปลี่ยนค่า 0 เป็น 1 หรือในทางกลับกันจาก 1 เป็น 0 ในปัจจุบันยีนอาจเป็นค่าจำนวนจริง รูปแบบการดัดแปลงยีนอาจมีได้ 3 แบบดังนี้ (Michalewicz, 1992)

(ก) Uniform Mutation ค่าของยีนถูกเปลี่ยนแปลงภายใต้ช่วงพิสัยที่กำหนด

(ข) Modified Uniform Mutation ค่าของยีนถูกเปลี่ยนแปลงโดยใช้ค่าคงที่เพียงค่าเดียว

(ค) Non-Uniform Mutation ค่าของยีนถูกเปลี่ยนแปลงโดยใช้ค่าที่ค่อย ๆ ลดลงเมื่อจำนวนรุ่น (Generation) เพิ่มมากขึ้น ซึ่งมีได้หลายวิธี สำหรับในการศึกษานี้ได้เลือกใช้วิธีการเปลี่ยนแปลงค่าของยีน โดยใช่ค่าจากการสุ่มที่มีการกระจายแบบ Gaussian หรือ Normal Distribution เรียกได้ชื่อวิธีดังกล่าวว่า Gaussian Mutation

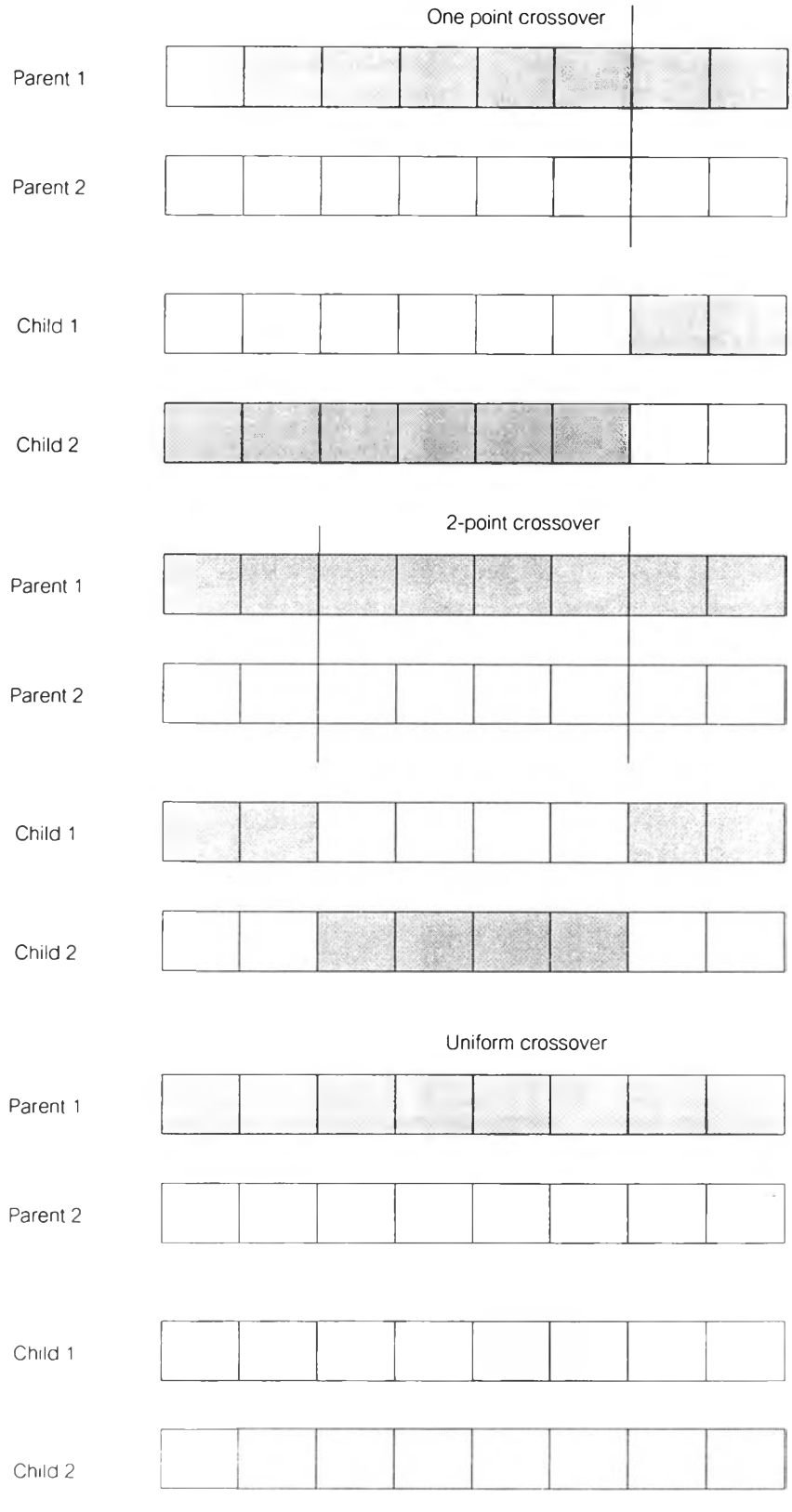
จากการศึกษาที่ผ่านมา พบว่า การใช้วิธีการปรับปรุงพันธุ์แบบ Gaussian Mutation มีความเหมาะสมและนิยมใช้ (Eiben และ Smith, 2003; Ursem, 2003)

3.2.3 พารามิเตอร์ของ GA (Parameter of GA)

(1) Probability of Crossover

Probability of Crossover (Pc) คือ ความน่าจะเป็นที่เกิดการผสมข้ามพันธุ์ระหว่าง โครโมโซมพ่อและแม่ โดยถ้าค่า Pc มีค่าเท่ากับ 100 เปอร์เซ็นต์ แปลว่าโครโมโซมใหม่ที่เกิดขึ้นมีการ ผสมข้ามพันธุ์ เกิดขึ้นทั้งหมด แต่ถ้า Pc มีค่าเท่ากับ 0 เปอร์เซ็นต์ แปลว่าโครโมโซมใหม่ที่เกิดขึ้นไม่มีการผสมข้ามพันธุ์เกิดขึ้น ทำให้ประชากรรุ่นใหม่ที่เกิดขึ้นนั้นไม่แตกต่างจากประชากรรุ่นเดิม เนื่องจากเป็นการคัดลอกจากประชากรรุ่นเดิมทั้งหมด

Pc เป็นตัวที่ใช้ควบคุมความถี่ของการ Crossover หากไม่เกิดการผสมข้ามพันธุ์ ขึ้น ทำให้โครโมโซมใหม่ (Offspring) ที่เกิดขึ้นเป็นการคัดลอกมาจากประชากรในรุ่นเดิม ถ้าเกิดมีการผสมข้ามพันธุ์ ขึ้นทำให้โครโมโซมใหม่ ที่เกิดขึ้นเป็นการคัดลอกบางส่วนมาจากพ่อและบางส่วนมาจากแม่ การผสมข้ามพันธุ์ ทำให้คาดได้ว่า โครโมโซมใหม่ที่เกิดขึ้นนั้นมียีน ที่มีค่าที่ดีของทั้งโครโมโซม พ่อและแม่ปนอยู่ด้วยกันซึ่งทำให้ โครโมโซมใหม่นั้นมีค่าที่ดีกว่า โครโมโซมเดิม ค่า Pc ที่เหมาะสมอยู่ในช่วง 0.5-0.9 (Goldberg, 1989; Agrawal และ Singh, 2003)



รูปที่ 3-7 วิธีการ Crossover

ที่มา : Wardlaw และ Sharif (1999)

(2) Probability of Mutation

Probability of Mutation (Pm) คือ ความน่าจะเป็นที่เกิดการปรับปรุงพันธุ์ขึ้นกับยีนใน โครโมโซม นั้น ๆ โดยถ้าค่า Pm มีค่าเท่ากับ 100 เปอร์เซ็นต์ แสดงว่าเกิดการปรับปรุงพันธุ์ขึ้นกับยีนทั้งหมดของทุก โครโมโซม ทำให้ค่าของยีนในโครโมโซม เป็นค่าที่ไม่ถูกต้อง (Fact Invert) แต่ถ้าไม่เกิดการปรับปรุงพันธุ์ขึ้น (Pm เท่ากับ 0 เปอร์เซ็นต์) ทำให้ได้ประชากรที่ไม่เปลี่ยนแปลง เพราะการเกิดปรับปรุงพันธุ์มีโอกาสที่อาจทำให้ได้ค่าที่ดีขึ้น

Pm เป็นตัวที่ใช้ควบคุมความถี่ของการปรับปรุงพันธุ์ หากไม่เกิดการปรับปรุงพันธุ์ขึ้น โครโมโซมใหม่ที่เกิดขึ้นภายหลังจากการเกิดผสมข้ามพันธุ์ กลายเป็นประชากรในรุ่นต่อไป แต่ถ้าหากมีการเกิดปรับปรุงพันธุ์ขึ้น โครโมโซมใหม่ที่เกิดขึ้นภายหลังจากการเกิด ผสมข้ามพันธุ์ มียีนบางส่วนถูกเปลี่ยนแปลงไปก่อนที่กลายเป็นประชากรในรุ่นต่อไป ค่า Pm ที่เหมาะสมอยู่ในช่วง 0.005-0.01 หรือ เท่ากับ 1/ความยาวโครโมโซม (Franchini และ Galeati, 1997; Obitko, 1998)

(3) ขนาดของประชากร (Population Size)

ขนาดของประชากรในแต่ละรุ่น (Generation) มีผลต่อการคัดเลือกหาค่าที่เหมาะสม เนื่องจากถ้าจำนวนประชากรมีน้อย จำนวนรอบการคำนวณมีมากขึ้นจนกว่าจะได้ชุดคำตอบที่เหมาะสมโดยรวม (Global Optimum) หรืออาจไม่พบเนื่องจากครบจำนวนรอบที่กำหนดไว้ แต่ในทางกลับกันหากมีจำนวนประชากรมาก ชุดคำตอบที่ให้ผลการคำนวณที่ดีที่สุดหรือผลการคำนวณที่เปลี่ยนแปลงน้อยมากเมื่อเปรียบเทียบระหว่างผลการคำนวณในแต่ละรุ่นที่อยู่ถัดไป ขนาดของจำนวนประชากรที่เพียงพอที่ให้ได้ค่าที่เหมาะสมมีค่าอยู่ในช่วง 30-100 (Goldberg, 1989)

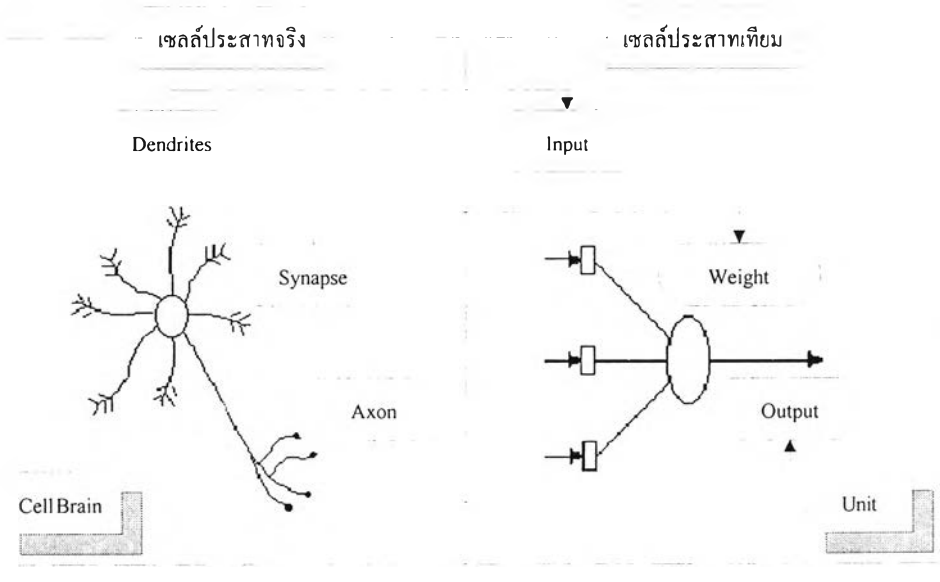
3.3 โครงข่ายใยประสาทเทียม (Artificial Neural Network, ANN)

โครงข่ายใยประสาทเทียมคือระบบประมวลข้อมูลที่มีการกระจายการคำนวณแบบขนานโดยเลียนแบบการทำงานของสมองมนุษย์ ซึ่งประกอบไปด้วยหน่วยประมวลผล (processing element) หลายๆเซลล์ ซึ่งทำหน้าที่คล้ายกับเซลล์สมองของมนุษย์ โดยที่แต่ละเซลล์จะโยงใยติดต่อกันโดยส่งสัญญาณเป็นผลลัพธ์ (Output) ผ่านส่วนที่เรียกว่า ไซแนพส์ (Synapse) กลายมาเป็นข้อมูลเข้า (Input) ของส่วนที่เรียกว่า เดนไดรต์ (Dendrites) และเมื่อผ่านกระบวนการประมวลผลจะได้ผลลัพธ์ออกมาในส่วนที่เรียกว่า แอ็กซอน (Axon) ในแต่ละเซลล์จะรับรู้ข้อมูลจากหลายทาง แล้วส่งต่อไปยังเซลล์อื่นๆ โดยใช้หลักการของการเชื่อมโยงเซลล์สมอง ดังรูปที่ 3-8 สำหรับความสัมพันธ์ระหว่างเซลล์ประสาทกับโครงข่ายใยประสาทเทียม แสดงดังตารางที่ 3-1

ตารางที่ 3-1 ความสัมพันธ์ระหว่างเซลล์ประสาทกับโครงข่ายใยประสาทเทียม

ลำดับ	เซลล์ประสาท	เซลล์ประสาทเทียม
1	ตัวเซลล์ (Cell body)	ยูนิต (unit)
2	เดนไดรต์ (Dendrites)	ตัวแปรเข้า (input)
3	แอ๊กซอน (Axon)	ตัวแปรออก (output)
4	ไซแนพส์ (Synapse)	ค่าน้ำหนัก (weight)
5	ความเร็วในการทำงานช้า	ความเร็วในการทำงานสูง
6	มีเซลล์จำนวนมาก (10^9 ยูนิต)	มีเซลล์จำนวนน้อยกว่า

ที่มา : เสรี ศุภราทิตย์ (2544)



รูปที่ 3-8 รูปเปรียบเทียบระหว่างเซลล์ประสาทกับโครงข่ายใยประสาทเทียม

ที่มา: เสรี ศุภราทิตย์ (2544)

หลักการทำงานของโครงข่ายใยประสาทเทียมจะมีแนวคิดที่แตกต่างกับแบบจำลองทั่วไปโดยสิ้นเชิง ในการจำลองพฤติกรรมโดยการใช้แบบจำลองโครงข่ายใยประสาทเทียม ไม่มีความจำเป็นที่จะต้องกำหนดหรือสร้างสมการแต่อย่างใด เพียงแต่เรารวบรวมข้อมูลนำเข้าและข้อมูลที่ต้องการ แต่ละเหตุการณ์ไว้เป็นคู่ๆ โดยทั่วไปการใช้แบบจำลองคณิตศาสตร์ หรือแบบจำลองทางสถิติจะต้องสร้างความสัมพันธ์ระหว่างตัวแปรเข้าและตัวแปรออก ซึ่งอยู่ในรูปของสมการแต่โครงข่ายใยประสาทเทียมจะทำการหาความสัมพันธ์ระหว่าง ข้อมูลนำเข้าและข้อมูลผลลัพธ์ โดยกระบวนการเรียนรู้จากข้อมูลจำนวนมากที่มีอยู่

การศึกษาครั้งนี้ ใช้โครงข่ายใยประสาทเทียมชนิดวิธีปรับแก้ค่าย้อนกลับ (Back Propagation Neural Network, BPNN) โดยโครงสร้างประกอบด้วย ชั้นนำเข้า (Input Layer) ชั้นแอบแฝง (Hidden Layer) และชั้นผลลัพธ์ (Output Layer) ในแต่ละชั้นประกอบด้วยโหนด (nodes) ซึ่งแต่ละโหนดของแต่ละชั้นนั้นจะถูกเชื่อมโยงด้วยค่าน้ำหนัก (weight) ลักษณะการทำงานจะทำงานแบบเคลื่อนที่ไปข้างหน้าอย่างเดียวแต่มีการปรับแก้ค่าน้ำหนักย้อนกลับซึ่งแสดงรายละเอียดในหัวข้อถัดไปและแสดงโครงสร้างของโครงข่ายใยประสาทเทียมดังรูปที่ 3-9

3.3.1 ขั้นตอนการเรียนรู้ภายในแบบจำลอง BPNN

สำหรับขั้นตอนการเรียนรู้ภายในแบบจำลอง BPNN นั้นประกอบด้วยสองขั้นตอนหลักคือ การคำนวณไปข้างหน้า (Forward pass) ซึ่งจะเป็นการประมวลข้อมูลจากชั้นนำเข้าเข้าสู่ชั้นผลลัพธ์ และขั้นตอนที่สองคือ การคำนวณย้อนกลับ (Backward pass) ซึ่งจะใช้ค่าผลต่างระหว่างค่าสังเกต (ข้อมูลจริง) และค่าคำนวณจากแบบจำลองในชั้นผลลัพธ์เป็นการปรับแก้ค่าความสัมพันธ์ระหว่างชั้น ดังแสดงในรูปที่ 3-9 โดยมีขั้นตอนดังต่อไปนี้

(1) แปลงข้อมูลนำเข้า และข้อมูลออก ให้อยู่ในรูปแบบมาตรฐานของค่าไร้หน่วย (Normalization) โดยในที่นี้จะมีค่าอยู่ระหว่าง 0.15-0.85 ดังสมการที่ 3-5

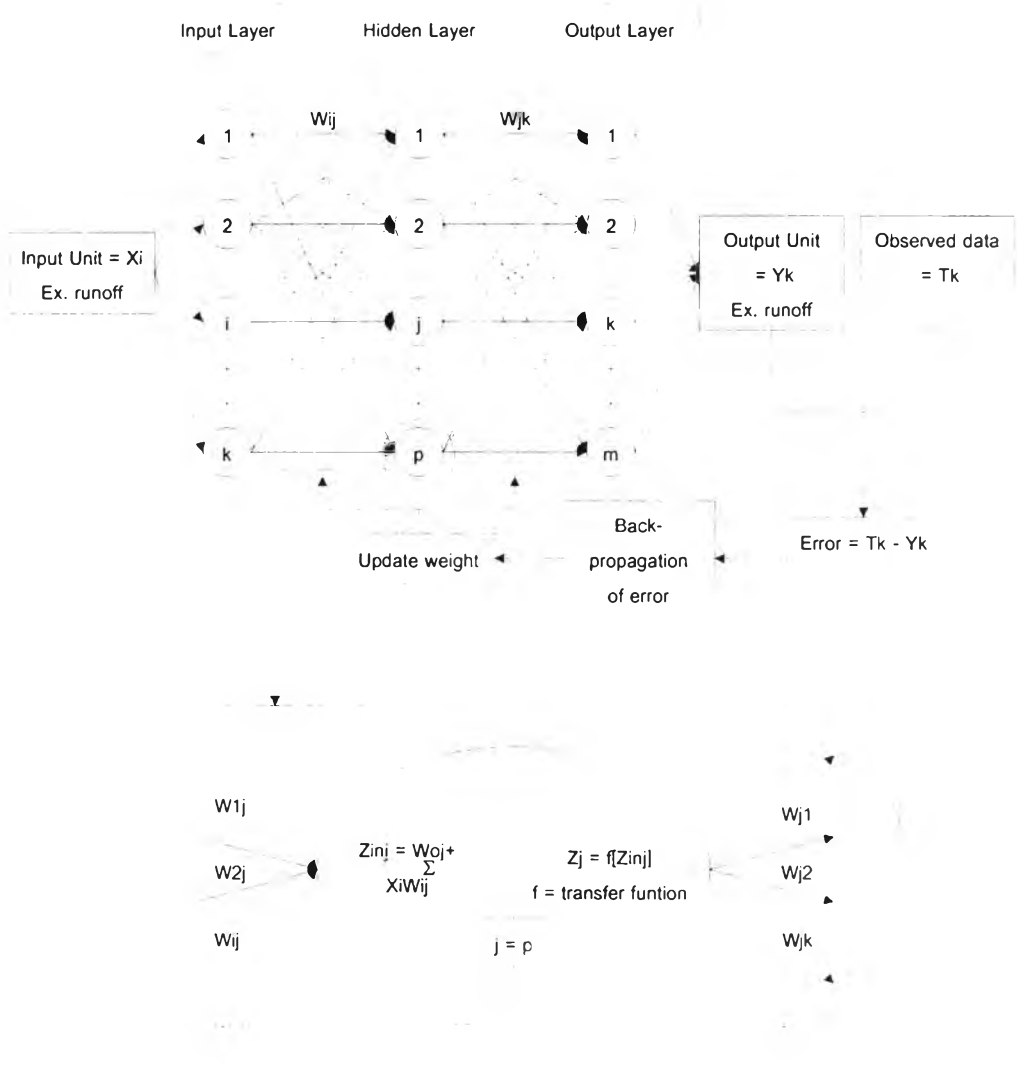
$$X_t' = \frac{0.70(X_t - \min)}{\max - \min} + 0.15 \qquad (3\text{-}5)$$

โดย X_t' คือ ค่าไร้หน่วยตัวที่ t

X_t คือ ค่าจริงของข้อมูลนำเข้าและข้อมูลผลลัพธ์ตัวที่ t

min คือ ค่าน้อยที่สุดของค่าจริง

max คือ ค่ามากที่สุดของค่าจริง



รูปที่ 3-9 ลักษณะโครงสร้างของแบบจำลอง BPNN และหลักการทำงานในหน่วยย่อย

ที่มา: เสรี ศุภราทิตย์ (2544)

(2) สมมุติค่าเริ่มต้นของค่าถ่วงน้ำหนักและค่าพารามิเตอร์ต่างๆในโครงสร้างแบบจำลอง

(3) กำหนดเงื่อนไขการหยุดการทำงานของกระบวนการเรียนรู้ เช่น ระดับความผิดพลาดที่ยอมรับได้ หรือจำนวนรอบการคำนวณ

การคำนวณไปข้างหน้า (Forward pass)

(4) ในชั้นนำเข้า (Input Layer) แต่ละโหนด (X_i, i = 1, 2,..., n) รับค่านำเข้า x_i และจะส่งต่อไปยังโหนดชั้นแอบแฝงซึ่งอยู่ถัดไป

(5) ในชั้นแอบแฝง (Hidden Layer) แต่ละโหนด (Z_j, j = 1, 2,..., n) รับค่าผลรวมของผลคูณระหว่างข้อมูลนำเข้า x_i และค่าน้ำหนักระหว่างชั้นนำเข้าและชั้นแอบแฝง W_{ij} ดังแสดงสมการที่3-6

$$Zin_j = w_{oj} + \sum x_i w_{ij} \quad (3\text{-}6)$$

เมื่อ Z_{inj} คือ ผลรวมของผลคูณระหว่างข้อมูลนำเข้า x_i และค่าน้ำหนักระหว่างชั้นนำเข้าและชั้นแอบแฝง W_{ij}

W_{oj} คือ ค่าปรับแก้ในชั้นแอบแฝง j

x_i คือ ค่าข้อมูลเข้าในชั้นนำเข้า

W_{ij} คือ ค่าน้ำหนักระหว่างหน่วย i ในชั้นนำเข้าและหน่วย j ในชั้นแอบแฝง

หลังจากนั้นค่า Zin_j จะถูกแปลงผ่านฟังก์ชั่นกระตุ้นแบบซิกมอยด์ ค่าที่ได้จะเป็นค่าผลลัพธ์จากชั้นแอบแฝงเพื่อส่งผ่านให้ชั้นต่อไป ดังแสดงสมการที่ 3-7

$$Z_j = f(Zin_j) \quad (3\text{-}7)$$

เมื่อ f คือ ฟังก์ชั่นกระตุ้นแบบซิกมอยด์ ดังแสดงในสมการที่ 3-8

$$f(Zin_j) = \frac{1}{1+\exp(-x)} \quad (3\text{-}8)$$

I21494495

(6) **ในชั้นผลลัพธ์** (Output Layer) คำนวณเช่นเดียวกับชั้นแอบแฝงในข้อ (5) โดยแต่ละโหนด (Y_k , k = 1, 2,..., n) รับค่าผลรวมของผลคูณระหว่างตัวแปรเข้า Z_j และค่าน้ำหนักระหว่างชั้นแอบแฝงและชั้นผลลัพธ์ W_{jk} ดังสมการที่ 3-9

$$Yin_k = w_{ok} + \sum z_i w_{jk} \quad (3\text{-}9)$$

เมื่อ Y_{ink} คือ ผลรวมของผลคูณระหว่างตัวแปรเข้า Z_j และค่าน้ำหนักระหว่างชั้นแอบแฝงและชั้นผลลัพธ์ W_{jk}

W_{ok} คือ ค่าปรับแก้ในชั้นแอบแฝง k

W_{jk} คือ ค่าน้ำหนักระหว่างโหนด j ในชั้นแอบแฝงและโหนด k ในชั้นผลลัพธ์

หลังจากนั้นค่า Yin_k จะถูกแปลงผ่านฟังก์ชั่นกระตุ้นแบบซิกมอยด์ ค่าที่ได้จะเป็นผลลัพธ์จากแบบจำลอง ดังสมการที่ 3-10

$$Y_k = f(Yin_k) \quad (3\text{-}10)$$

การคำนวณย้อนกลับ (Backward pass)

(7) เมื่อคำนวณถึงชั้นผลลัพธ์ ค่าผลลัพธ์จากแบบจำลองแต่ละโหนดในชั้นผลลัพธ์จะถูกนำไปคำนวณพจน์ความผิดพลาด (δ_k) โดยเทียบกับข้อมูลจริง ดังสมการที่ 3-11

$$\delta_k = (T_k - y_k) f'(Yin_k) \quad (3\text{-}11)$$

เมื่อ T_k คือ ค่าที่ได้จากข้อมูลจริง

$f'(\ )$ คือ อนุพันธ์ของฟังก์ชั่นกระตุ้นแบบซิกมอยด์ ดังสมการที่ 3-12

$$f'(Yin_k) = f(Yin_k) \times (1 - f(Yin_k)) \quad (3\text{-}12)$$

เมื่อได้พจน์ความผิดพลาด (δ_k) แล้วจึงทำการคำนวณค่าปรับแก้น้ำหนักระหว่างชั้นผลลัพธ์และชั้นแอบแฝง $\Delta w_{jk}(n+1)$ เพื่อใช้ปรับแก้ค่า $w_{jk}(\ )$ ดังสมการที่ 3-13

$$\Delta w_{jk}(n+1) = \eta \delta_k Y_k + \alpha \Delta w_{jk}(n) \quad (3\text{-}13)$$

เมื่อ η คือ ค่าอัตราการเรียนรู้ มีค่าระหว่าง 0 ถึง 1

α คือ ค่าโมเมนตัม มีค่าระหว่าง 0 ถึง

$\Delta w_{jk}(n)$ คือ ค่าปรับแก้น้ำหนักระหว่างโหนด j ในชั้นตัวแอบแฝงและโหนด k ในชั้นผลลัพธ์ ในรอบที่ n (ในรอบที่ n = 1 มีค่าเท่ากับ 0)

(8) สำหรับชั้นแอบแฝงคำนวณพจน์ความผิดพลาด (δ_j) เช่นเดียวกับชั้นผลลัพธ์ ดังสมการที่ 3-14

$$\delta_j = \sum \delta_k w_{jk} \; f'(Zin_j) \tag{3-14}$$

นำค่าพจน์ความผิดพลาด (δ_j) ที่ได้คำนวณค่าปรับแก้น้ำหนักระหว่างชั้นนำเข้าและชั้นแอบแฝง $\Delta w_{ij}(n+1)$ เพื่อใช้ปรับแก้ค่า $w_{ij}(\;)$ ดังสมการที่ 3-15

$$\Delta w_{ij}(n+1) = \eta \delta_j Z_j + \alpha \Delta w_{ij}(n) \tag{3-15}$$

$\Delta w_{ij}(n)$ คือ ค่าปรับแก้น้ำหนักระหว่างโหนด i ในชั้นนำเข้าและโหนด j ในชั้นแอบแฝง ในรอบที่ n (ในรอบที่ n = 1 มีค่าเท่ากับ 0)

(9) สำหรับแต่ละโหนด (Y_k, k = 1, 2,..., n) ในชั้นผลลัพธ์ทำการปรับแก้ค่าน้ำหนักเพื่อใช้ในการคำนวณใหม่ในรอบที่ n+1 ดังสมการที่ 3-16

$$w_{jk}(n+1) = w_{jk}(n) + \Delta w_{jk}(n+1) \tag{3-16}$$

สำหรับแต่ละโหนด (Z_j, j = 1, 2,..., n) ในชั้นตัวแอบแฝงทำการปรับแก้ค่าน้ำหนักเพื่อใช้ในการคำนวณใหม่ในรอบที่ n+1 เช่นเดียวกัน ดังสมการที่ 3-17

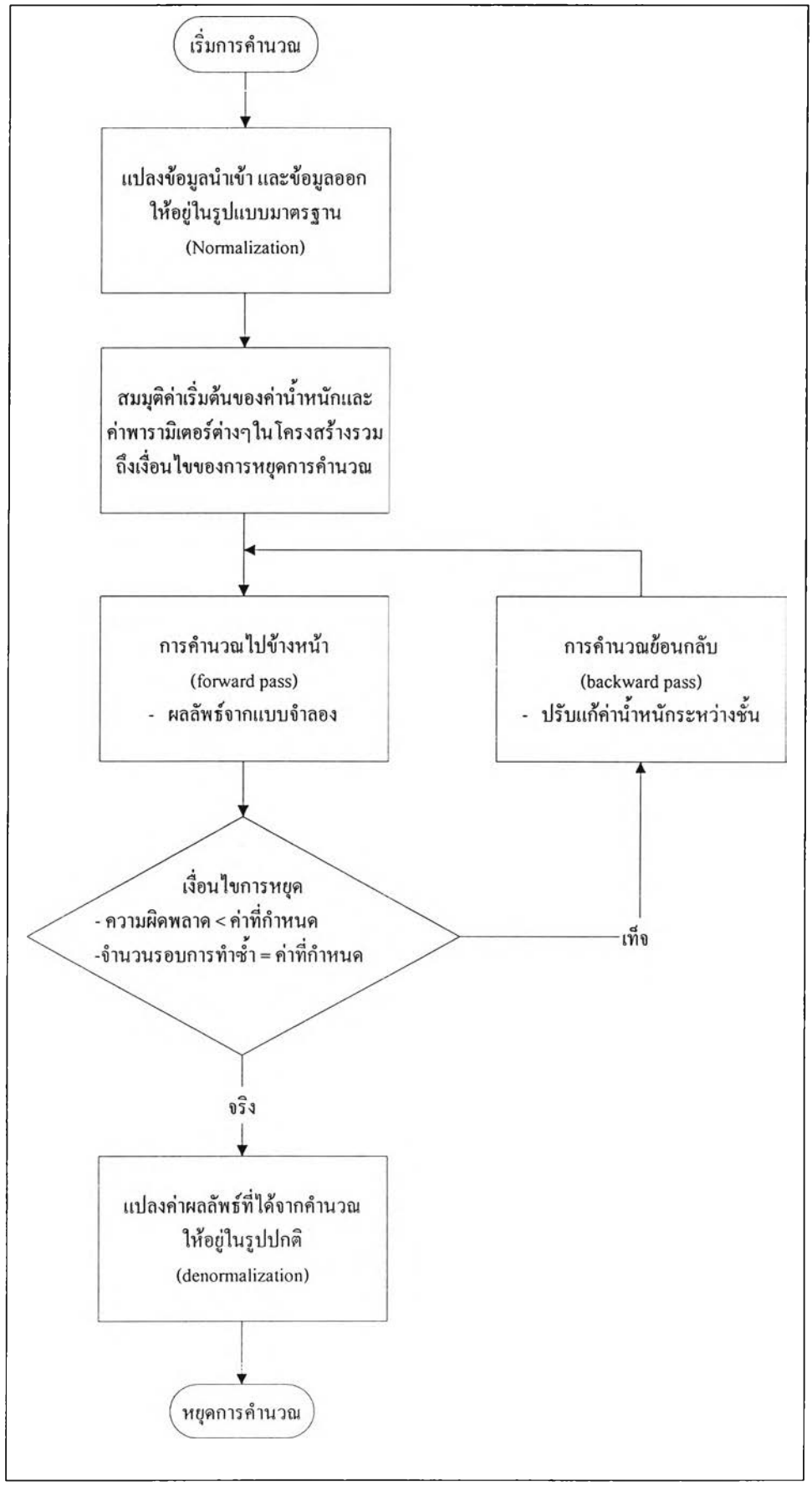
$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n+1) \tag{3-17}$$

(10) ทำการคำนวณใหม่ตั้งแต่ข้อที่ (5)- (9) และจะหยุดการทำซ้ำตามเงื่อนไขการหยุดการทำงานของกระบวนการเรียนรู้ในข้อที่ (3)

(11) แปลงค่าผลลัพธ์ที่ได้จากคำนวณ Y_k กลับให้อยู่ในรูปปกติที่มีหน่วยเช่นเดียวกับข้อมูลของผลลัพธ์ (Denormalization) ดังสมการที่ 3-18

$$X_i = \frac{(\max - \min)(Yk - 0.15)}{0.7} + \min \tag{3-18}$$

จากขั้นตอนการคำนวณทั้งหมดสามารถสรุปดังรูปที่ 3-10



รูปที่ 3-10 ผังการไหลแสดงการทำงานของการเรียนรู้ของแบบจำลอง BPNN

3.3.1 ตัวแปรและองค์ประกอบภายในแบบจำลอง BPNN

การพิจารณาตัวแปรและองค์ประกอบภายในแบบจำลอง BPNN ซึ่งอาจส่งผลต่ออัตราการลู่เข้าหรือความถูกต้อง ประกอบด้วย

(1) ค่าถ่วงน้ำหนักเริ่มต้น

วิธีการหาค่าถ่วงน้ำหนักเริ่มต้นของแบบจำลอง BPNN เป็นวิธีการสุ่มค่า (random) ด้วยฟังก์ชั่นการสุ่มทางคณิตศาสตร์ โดยค่าสุ่มจะอยู่ในช่วงของค่าสูงสุด-ต่ำสุดที่กำหนดไว้ ซึ่งจากการทบทวนผลการศึกษาที่ผ่านมาที่ได้แสดงการเลือกใช้ค่าสูงสุด-ต่ำสุดของค่าถ่วงน้ำหนักเริ่มต้นของ BPNN ดังนี้

ASCE (2000) กล่าวว่า ค่าถ่วงน้ำหนักเริ่มต้นที่ดีจะทำให้การเรียนรู้ของ BPNN เร็วขึ้น อย่างไรก็ตามไม่สามารถที่จะกำหนดค่าสุ่มเริ่มต้นที่ดีๆได้ ดังนั้นขนาดของค่าถ่วงเริ่มต้นควรจะมีค่าต่ำให้อยู่ในช่วง (-0.3, 0.3)

Thriumalaiah และ Deo (1998) ได้กำหนดช่วงพิสัยของค่าถ่วงน้ำหนักเริ่มต้นในการพยากรณ์น้ำท่าและระดับน้ำไว้เท่ากับ (-0.5, 0.5)

เสรี ศุภราทิตย์ (2544) กล่าวว่า การเลือกค่าถ่วงน้ำหนักให้มีค่าต่ำซึ่งอยู่ระหว่าง -1 ถึง 1 ช่วยหลีกเลี่ยงปัญหาการไม่ลู่เข้าหาคำตอบสุดท้าย (divergence) ได้ เพราะการเลือกค่าถ่วงน้ำหนักมากๆจะทำให้ค่าอนุพันธ์ของฟังก์ชั่นกระตุ้นเข้าใกล้ศูนย์

Huang และคณะ (2004) ได้กำหนดช่วงพิสัยของค่าถ่วงน้ำหนักเริ่มต้นในการพยากรณ์ไว้เท่ากับ (0.1, 1.0)

ดังนั้นในการศึกษานี้ได้พัฒนาโปรแกรม BPNN ที่มีช่วงพิสัยของการสุ่มค่าถ่วงน้ำหนักเริ่มต้นอยู่ในช่วง (-0.5, 0.5)

(2) จำนวนหน่วยในชั้นแอบแฝงและจำนวนชั้นแอบแฝง

ในช่วงแรกยังไม่มีแนวทางการกำหนดโครงสร้างในชั้นแอบแฝงที่ชัดเจน ส่วนใหญ่ใช้การลองผิดลองถูก (ASCE, 2000) ซึ่งกินเวลานาน เพื่อหาโครงสร้างที่เหมาะสม ต่อมาจึงมีการกำหนดโครงสร้างให้สัมพันธ์กับจำนวนหน่วยในชั้นตัวแปรเข้า เช่น จำนวนหน่วยในชั้นแอบแฝงเป็นจำนวนเท่าของจำนวนหน่วยในชั้นนำเข้า และต่อมามีการใช้ทฤษฎีการหาค่าที่เหมาะสมมาประยุกต์ใช้ เช่น Genetic Algorithm (เสรี ศุภราทิตย์, 2001)

(3) ฟังก์ชั่นกระตุ้น (Activation Function)

ฟังก์ชั่นกระตุ้นที่นิยมใช้โดยทั่วไปมีอยู่หลายแบบทั้งแบบเชิงเส้นและไม่เชิงเส้น โดยคุณสมบัติของฟังก์ชั่นกระตุ้นจะต้องอยู่ในขอบเขตและสามารถหาอนุพันธ์ได้ โดยฟังก์ชั่นซิกมอยด์เป็นที่นิยมมากที่สุดเนื่องจากมีความสะดวกในการหาอนุพันธ์ ดังแสดงในสมการที่ 3-8

(4) อัตราการเรียนรู้ (learning rate, η)

อัตราการเรียนรู้จะเป็นตัวแปรหนึ่งในการกำหนดขนาดการปรับค่าถ่วงน้ำหนัก ดังสมการที่ 3-13 และ 3-15 โดยทั่วไปกำหนดให้เป็นค่าคงที่ระหว่าง 0.05 - 0.90 หากเลือกค่าที่สูงเกินไปจะทำให้การเดินทางไปหาความคลาดเคลื่อนที่ต่ำที่สุดมีการแกว่งและอาจไม่มีการลู่เข้าหาคำตอบที่ต้องการ ในทางตรงข้ามการเลือกค่าอัตราการเรียนรู้ที่น้อยเกินไปก็เป็นการสิ้นเปลืองเวลาในการคำนวณ (พงษ์ศักดิ์ สุทธินนท์, 2546)

(5) ค่าโมเมนตัม (momentum, α)

ในกระบวนการเรียนรู้ ค่าโมเมนตัมอาจจะใช้หรือไม่ใช้ก็ได้ อย่างไรก็ตามการเพิ่มค่าโมเมนตัมจะช่วยป้องกันการแกว่งของระบบ การเพิ่มค่าโมเมนตัมเป็นการเพิ่มสัดส่วนน้ำหนักของค่าถ่วงน้ำหนัก ดังสมการที่ 3-13 และ 3-15 การเพิ่มสัดส่วนดังกล่าวจะช่วยป้องกันการเปลี่ยนแปลงค่าถ่วงน้ำหนักที่รุนแรงหรือผิดปกติ ค่าโมเมนตัมจะเป็นค่าบวกที่น้อยกว่า 1 โดยทั่วไปมีค่าระหว่าง 0.5 ถึง 0.9 (พงษ์ศักดิ์ สุทธินนท์, 2546)

3.4 ค่าสถิติที่ใช้วัดประสิทธิภาพของแบบจำลอง

วัตถุประสงค์หลักในการพยากรณ์ คือ การให้ผลการพยากรณ์มีความถูกต้องใกล้เคียงกับค่าจริง ในทางสถิติตัวแปรสถิติที่นิยมใช้วัดประสิทธิภาพและค่าความคลาดเคลื่อนในการคำนวณของแบบจำลองมีดังนี้

(1) ดรรชนีวัดประสิทธิภาพ (Efficiency Index, EI)

$$EI = \frac{\sum_{i=1}^{N}(Qi-\bar{Q})^2 - \sum_{1}^{N}(Qi-Fi)^2}{\sum_{i=1}^{N}(Qi-\bar{Q})^2} = \frac{SST-SSE}{SST} \qquad (3\text{-}19)$$

เมื่อ SST คือ ค่าความแปรปรวนทั้งหมดในตัวแปร Q

SSE คือ ผลรวมของผลต่างยกกำลังสองระหว่าง ตัวแปร Q และ F

Qi คือ ค่าจริงหรือค่าเป้าหมายของตัวแปร Q ณ เวลา i

$\overline{Q}$ คือ ค่าเฉลี่ยของตัวแปร Q โดย $\overline{Q} = \frac{1}{N}\sum_{i=1}^{N} Qi$

Fi คือ ผลการคำนวณที่ได้ ณ เวลา i

N คือ จำนวนข้อมูลทั้งหมด

(2) ค่าเบี่ยงเบนมาตรฐานของค่าคลาดเคลื่อน (Standard error, S_e) หรือ ค่ารากที่สองของค่า คลาดเคลื่อนกำลังสองเฉลี่ย (Root Mean Square Error, RMSE)

$$S_e = RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Qi - Fi)^2} \qquad (3\text{-}20)$$

(3) ค่าคลาดเคลื่อนเฉลี่ยสัมบูรณ์ (Mean Absolute Deviation, MAD)

$$MAD = \frac{1}{N}\sum_{i=1}^{N}|Qi - Fi| \qquad (3\text{-}21)$$

(4) เปอร์เซ็นต์ค่าความคลาดเคลื่อนสัมพัทธ์เฉลี่ยสัมบูรณ์ (Average Absolute Relative Error, AARE)

$$AARE = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|Qi - Fi|}{Qi}\right) \times 100 \qquad (3\text{-}22)$$

(5) ค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation coefficient, r)

$$r = \frac{n\sum XiYi - \left(\sum Xi\right)\left(\sum Yi\right)}{\sqrt{\left[n\sum X^2 i - \left(\sum Xi\right)^2\right]\left[nY^2 i - \left(\sum Yi\right)^2\right]}} \qquad (3\text{-}23)$$

ค่าสัมประสิทธิ์สหสัมพัทธ์ r คือ ดัชนีที่ใช้วัดทิศทางและระดับความสัมพันธ์เชิงเส้นระหว่างตัวแปร X และ Y โดยมีค่าอยู่ระหว่าง -1 กับ 1 เครื่องหมายบวก ลบของค่า r จะแสดงทิศทางของความสัมพันธ์ระหว่าง 2 ตัวแปร หากว่า ค่า r เป็นบวก หมายถึง เมื่อ X เพิ่ม Y จะเพิ่มตามไปด้วย ในทางตรงข้าม หากค่า r เป็นลบ หมายถึง เมื่อ X กับ Y จะมีความสัมพันธ์ในลักษณะที่ตรงกันข้าม หรือ อาจแสดงในรูปของ R^2 ซึ่งเท่ากับ ค่า r ยกกำลังสอง