

จินเนติกอัลกอริทึม Genetic Algorithm

จินเนติกอัลกอริทึม (genetic algorithm) เป็นวิธีการหาคำตอบที่ดีที่สุดโดยใช้หลักการคัดเลือกแบบธรรมชาติ และหลักการทางสายพันธุ์ จินเนติกอัลกอริทึมเป็นการคำนวณอย่างหนึ่งที่สามารถกล่าวได้ว่ามี “วิวัฒนาการ” อยู่ในขั้นตอนของการหาคำตอบ และได้รับการจัดให้เป็นวิธีหนึ่งในกลุ่มของการคำนวณเชิงวิวัฒนาการ (evolutionary computing) ซึ่งปัจจุบันเป็นที่ยอมรับในประสิทธิภาพ และมีการนำไปประยุกต์ใช้อย่างกว้างขวางในการแก้ปัญหาการหาค่าเหมาะที่สุด เนื้อหาในหัวข้อนี้นำเสนอรายละเอียดขององค์ประกอบและโครงสร้างของจินเนติกอัลกอริทึม รวมไปถึงตัวอย่างการประยุกต์ใช้งานแบบต่างๆ เพื่อความเข้าใจและสามารถนำไปออกแบบประยุกต์ใช้งานได้จริง

จินเนติกอัลกอริทึมหรือเรียกโดยย่อว่า “GA” ถูกพัฒนาขึ้นในช่วงทศวรรษที่ 60 โดยจำลองเอาแนวคิดของการวิวัฒนาการของสิ่งมีชีวิตในระบบชีววิทยามาใช้ในการคำนวณด้วยคอมพิวเตอร์ GA ได้รับความนิยมและเป็นที่ยอมรับอย่างกว้างขวาง โดยการเผยแพร่ของ John Holland [Holland, 1975][Goldberg, 1989][Goldberg, 1994] ในหนังสือชื่อ "Adaptation in Natural and Artificial Systems" หนังสือดังกล่าวได้ตีพิมพ์เป็นครั้งแรกในปี ค.ศ. 1975 หลังจากนั้นจึงมีการนำเอา GA ไปประยุกต์ใช้งานในด้านต่างๆ กันอย่างแพร่หลาย พร้อมๆ กับการศึกษาและพัฒนาองค์ประกอบต่างๆ ของ GA ให้มีประสิทธิภาพดียิ่งขึ้น GA ถือว่าเป็นวิธีการหาคำตอบที่เหมาะสมที่สุดเชิงผสมผสาน (combinatorial optimization) แบบปัญญาเชิงคำนวณ ที่มีความสามารถในการหาคำตอบอย่างชาญฉลาด และลดความยุ่งยากในขั้นตอนต่างๆ ของการค้นหาลงไป ซึ่งวิธีการหาคำตอบดังกล่าวนี้มีข้อได้เปรียบและมีความแตกต่างไปจากวิธีดั้งเดิม เช่น การโปรแกรมเชิงเส้น วิธีซิมเพล็กซ์ หรือวิธีการค้นหาผลเฉลยรูปแบบปิด ปัจจุบันจึงเห็นได้ว่าการนำเอา GA ไปใช้ในเกือบจะทุกสาขาวิชา ตัวอย่างบางส่วนที่น่าสนใจล่าสุด เช่น

- การประมวลผลสัญญาณดิจิทัล ดู [Man et al., 1997][Castillo et al., 2001]
- การประมวลผลสัญญาณภาพและการมองเห็นด้วยคอมพิวเตอร์ ดู [Minami et al., 2001][Bosco, 2001][Hussein et al., 2001][Mitsukura et al., 2001]
- ระบบควบคุม ดู [Man et al., 1997][Bedwani and Ismail, 2001][Visioli, 2001][Melin and Castillo, 2001]

- การสื่อสารและโทรคมนาคม ดู [Bajwa et al., 2001][Weile and Michielssen, 2001][Arabas and Kozdrowski, 2001]
- อิเล็กทรอนิกส์ ดู [Grimbleby, 2000][Manganaro, 2000][Goh and Li, 2001]
- ไฟฟ้ากำลัง ดู [Wong et al., 2000][Poirier et al., 2001][Yong-Hua and Irving, 2001][Kozunovic and Liao, 2001]
- คอมพิวเตอร์และอินเทอร์เน็ต ดู [Nick and Themis, 2001][Min-Huang et al., 2001][Kim and Byoung-Tak, 2001]
- การแพทย์ ดู [Meesad and Yen, 2001][Moller and Zeipelt, 2001][Kin et al., 2001]
- การเงิน ดู [Lam, 2001]
- การขนส่ง ดู [Srinivasan et al., 2001]

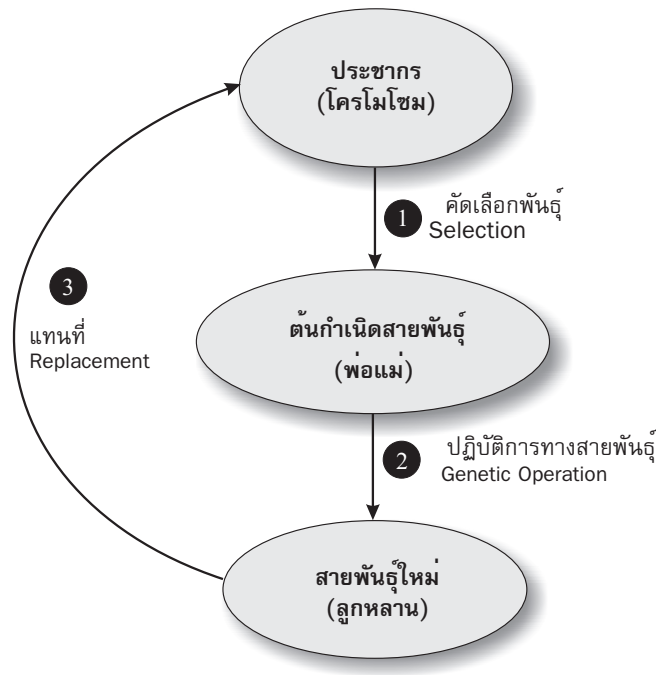
นอกจากนั้นแล้ว GA ยังถูกนำไปใช้ร่วมกับเครื่องมืออื่นๆ ได้อย่างมีประสิทธิภาพ เช่น ระบบฟัซซี่ลอจิก [Cor-don et al., 2001] ระบบที่ใช้เวฟเล็ท [Jones et al., 2001] ระบบเครือข่ายประสาทเทียม [Yamazaki et al., 1998] เป็นต้น จะเห็นได้ว่าประสิทธิภาพและสมรรถนะของ GA ได้เป็นที่ยอมรับและมีการนำไปประยุกต์ใช้กันอย่างแพร่หลาย ในหัวข้อต่อไปจะได้นำเสนอรายละเอียดของ GA ในทุกองค์ประกอบเพื่อความเข้าใจในหลักการทำงานของ GA อันจะนำไปสู่การนำไปประยุกต์ใช้ได้ต่อไปอย่างมีประสิทธิภาพ

3.1 GA คืออะไร?

GA เป็นขั้นตอนในการค้นหาคำตอบให้กับระบบ เราสามารถมอง GA เป็นเครื่องมือในการช่วยคำนวณอย่างหนึ่ง ที่คอยทำงานอยู่ข้างๆ ระบบของเรา วัฏจักรของ GA โดยธรรมชาติแล้วประกอบไปด้วย 3 กระบวนการที่สำคัญ ได้แก่ (ดังแสดงในรูปที่ 3.1)

1. **การคัดเลือกสายพันธุ์ (selection)** คือขั้นตอนในการคัดเลือกประชากรที่ดีในระบบ ไปเป็นต้นกำเนิดสายพันธุ์เพื่อให้กำเนิดลูกหลานในรุ่นถัดไป
2. **ปฏิบัติการทางสายพันธุ์ (genetic operation)** คือกรรมวิธีการเปลี่ยนแปลงโครโมโซมด้วยวิธีการทางสายพันธุ์ เป็นขั้นตอนการสร้างลูกหลาน ซึ่งได้จากการรวมพันธุ์ของต้นกำเนิดสายพันธุ์เพื่อให้ได้ลูกหลานที่มีส่วนผสมผสานมาจากพ่อแม่ หรือได้จากการแปรผกผันของพ่อแม่เพื่อให้ได้ลูกหลานสายพันธุ์ใหม่เกิดขึ้น
3. **การแทนที่ (replacement)** คือขั้นตอนการนำเอาลูกหลานกำเนิดใหม่ไปแทนที่ประชากรเก่าในรุ่นก่อน เป็นกระบวนการในการคัดเลือกกว่าควรจะเอาลูกหลานในกลุ่มใด จำนวนเท่าไร ไปแทนประชากรเก่าในกลุ่มใด

วัฏจักรของ GA แสดงให้เห็นถึงความเหมือนกับการอยู่รอดของสิ่งมีชีวิตในธรรมชาติ สิ่งมีชีวิตที่มีการปรับตัวให้เข้ากับสภาพแวดล้อมได้ดีกว่าจะสามารถอยู่รอดได้ ในขณะที่สิ่งมีชีวิตอื่นๆ ที่ไม่สามารถปรับตัวเองได้จะต้องสูญพันธุ์ไป การปรับตัวดังกล่าวแสดงว่าสิ่งมีชีวิตนั้นมีวิวัฒนาการเกิดขึ้น ไดโนเสาร์ต้องสูญพันธุ์ไปจากโลกนี้ก็เนื่องมาจากการไม่มีวิวัฒนาการที่เพียงพอต่อการอยู่รอด ในขณะที่สิ่งมีชีวิตหลายๆ ชนิดในปัจจุบันที่วิวัฒนาการมาจากไดโนเสาร์ และมีการปรับตัวให้อยู่ในสภาพแวดล้อมที่เปลี่ยนแปลงไป จะสามารถรักษาเผ่าพันธุ์อยู่มาจนถึงปัจจุบันได้ GA มีการจำลองวิวัฒนาการของสิ่งมีชีวิตในระบบธรรมชาติ กล่าวคือกระบวนการภายในของ GA ทำให้คำตอบของระบบที่มีอยู่เกิดวิวัฒนาการในตัวเองอันจะนำไปสู่การปรับตัวให้กลายเป็นคำตอบที่ดีกว่าและดีที่สุดได้ รายละเอียดต่างๆ ขององค์ประกอบในวัฏจักร GA มีดังนี้



รูปที่ 3.1: วัฏจักรของ GA

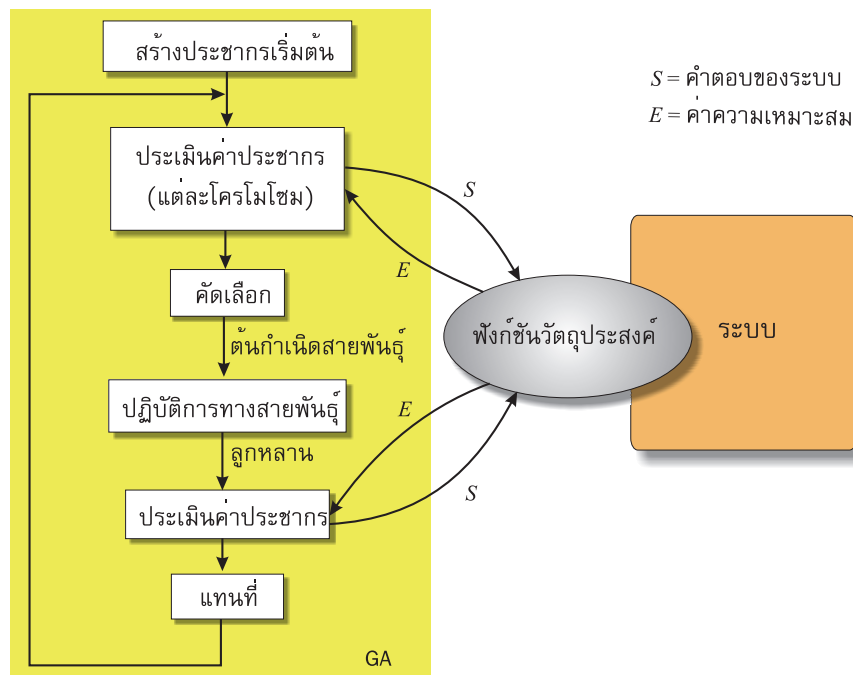
- **ประชากร (population)** - ประกอบไปด้วยกลุ่มของโครโมโซม (chromosome) ซึ่งเป็นตัวแทนของคำตอบในระบบที่ต้องการค้นหา
- **ต้นกำเนิดสายพันธุ์ (parents)** - กลุ่มประชากรที่ถูกคัดเลือกเพื่อเป็นตัวแทนในการให้กำเนิดสายพันธุ์ใหม่ในรุ่นถัดไป (next generation) ประชากรกลุ่มนี้จะเปรียบเสมือนกับเป็น ‘พ่อแม่’ สำหรับใช้ในการสืบทอดสายพันธุ์ให้ลูกหลานต่อไป
- **สายพันธุ์ใหม่ (offspring)** - หรือ ‘ลูกหลาน’ เป็นประชากรกลุ่มใหม่ที่ได้รับการถ่ายทอดสายพันธุ์มาจากพ่อแม่ โดยคาดหวังที่จะได้รับสายพันธุ์ที่ดีที่สุดเพื่อถ่ายทอดต่อไป กันในประชากรรุ่นถัดไป

3.2 ขั้นตอนการทำงานของ GA

แผนภาพในรูปที่ 3.2 แสดงขั้นตอนทั่วไปของ GA และการเชื่อมโยงเข้ากับระบบในโลกจริงเพื่อทำการค้นหาคำตอบที่ต้องการ คำตอบของระบบที่ต้องการให้ GA ทำการค้นหาคำตอบจะอยู่ในรูปของโครโมโซมในกลุ่มของประชากร (แน่นอนว่าคำตอบที่ต้องการจะต้องเป็นโครโมโซมที่ดีที่สุดในกลุ่ม) ดังนั้นระบบจะสามารถรู้ได้ว่าคำตอบที่มีอยู่ใน GA ณ เวลาหนึ่งๆ นั้นดีหรือไม่ดีอย่างไรด้วยการประเมินค่าของโครโมโซม ระบบจะมีการเชื่อมต่อกับ GA ผ่านฟังก์ชันวัตถุประสงค์สำหรับการประเมินค่าของโครโมโซมซึ่งจะมีรายละเอียดในหัวข้อต่อไป

ขั้นตอนการทำงานทั่วไปของ GA

1. สร้างประชากร โดยปกติจะใช้การสุ่ม (random)
2. ประเมินค่าโครโมโซมของกลุ่มประชากรทั้งหมดด้วยฟังก์ชันวัตถุประสงค์ เนื่องจากระบบไม่สามารถเข้าใจค่าของโครโมโซมภายใน GA ดังนั้นโครโมโซมจะต้องผ่านการถอดรหัสก่อนที่จะนำไปทำการคำนวณด้วยฟังก์ชันวัตถุประสงค์ได้

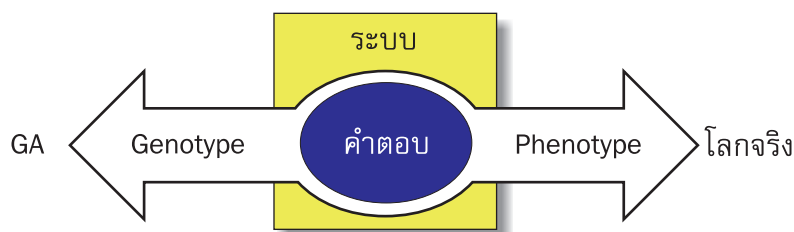


รูปที่ 3.2: ขั้นตอนทั่วไปของ GA กับการเชื่อมโยงเข้ากับระบบ

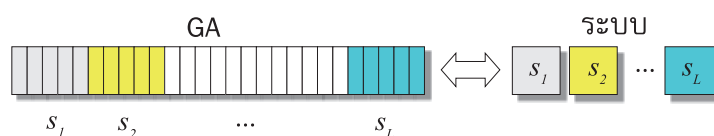
3. คำนวณค่าความเหมาะสมแล้วส่งกลับไปยัง GA
4. ใช้ค่าความเหมาะสมทำการคัดเลือกโครโมโซมบางกลุ่ม เพื่อนำมาเป็นต้นกำเนิดสายพันธุ์ ซึ่งจะถูกใช้เป็นตัวแทนในการถ่ายทอดสายพันธุ์ให้กับรุ่นถัดไป
5. นำต้นกำเนิดสายพันธุ์มาทำการสร้างลูกหลาน ด้วยปฏิบัติการทางสายพันธุ์ โครโมโซมที่ได้ในขั้นตอนนี้ก็คือโครโมโซมลูกหลาน
6. คำนวณค่าความเหมาะสมของโครโมโซมลูกหลาน โดยใช้ขั้นตอนเดียวกับข้อ 3)
7. โครโมโซมในประชากรเดิมจะถูกแทนที่ด้วยลูกหลานที่ได้จากข้อ 5) ประชากรเพียงบางส่วนเท่านั้นที่จะถูกแทนที่ ด้วยกลวิธีเฉพาะสำหรับขั้นตอนของการแทนที่โดยใช้ค่าความเหมาะสมในการตัดสินใจ
8. เริ่มต้นทำซ้ำจากขั้นตอนในข้อ 2) ไปเรื่อยๆ จนกระทั่งได้คำตอบที่ต้องการ คำตอบที่ได้จะมาจากโครโมโซมที่ดีที่สุดในกลุ่มประชากรนั่นเอง โดยที่สามารถใช้ค่าจากฟังก์ชันวัตถุประสงค์เพื่อเป็นการประเมินว่าคำตอบที่ได้เป็นที่ต้องการแล้วหรือไม่

จากกระบวนการของ GA จะเห็นได้ว่าสิ่งที่จำเป็นจะต้องทำการออกแบบเพื่อนำ GA มาใช้ก็คือ

- โครโมโซมประชากรและการเข้ารหัส
- การประเมินค่าความเหมาะสม
- การคัดเลือกต้นกำเนิดสายพันธุ์
- ปฏิบัติการทางสายพันธุ์
- การแทนที่



รูปที่ 3.3: คำตอบในมุมมองของ GA และโลกจริง



รูปที่ 3.4: โครโมโซมที่ถูกเข้ารหัสแบบฐานสองจากคำตอบในระบบ

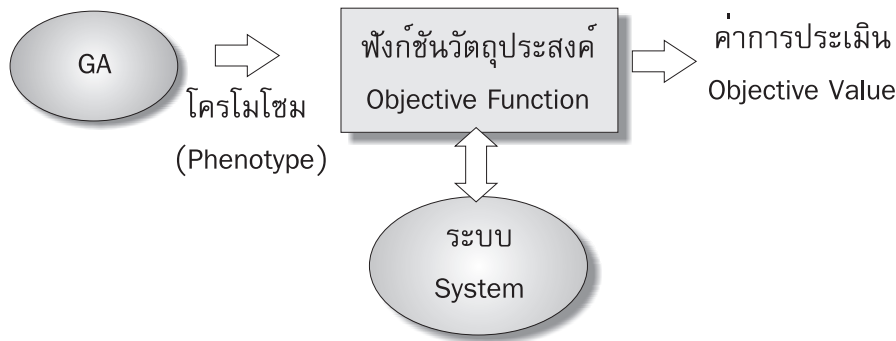
3.3 โครโมโซมประชากรและการเข้ารหัส Population & Encoding Scheme

GA พิจารณาคำตอบของปัญหาจากกลุ่มของคำตอบหรือประชากร (population) ของคำตอบ แต่ละคำตอบจะมีคุณลักษณะเฉพาะตัวซึ่งแสดงอยู่ในรูปของโครโมโซม (chromosome) หรือในรูปจีโนม (genome) การเข้ารหัสประชากรเป็นขั้นตอนแรกและเป็นขั้นตอนที่สำคัญ เพราะเป็นการออกแบบให้โครโมโซมเป็นตัวแทนของคำตอบจากระบบ ในการใช้งาน GA ทั่วๆ รูปแบบที่ง่ายที่สุดคือการกำหนดให้โครโมโซมอยู่ในรูปของตัวแปรแบบสตริง (string of variables)

$$S = [s_1, s_2, \dots, s_L] \quad (3.1)$$

โดยที่ S คือโครโมโซมหนึ่งๆ และแต่ละ $s_i, i = 1, 2, \dots, L$ คือแต่ละตัวแปรในชุดคำตอบของระบบ (แต่ละระบบจะมีจำนวนตัวแปรไม่เท่ากัน ขึ้นอยู่กับลักษณะของปัญหา ความซับซ้อนและการออกแบบการแก้ปัญหาของระบบนั้นๆ)

เมื่อพิจารณาโครโมโซมหนึ่งๆ เราสามารถมองได้ว่าเป็นการนำเอาคำตอบทั้งชุดของระบบมาวางเรียงต่อกันเป็นสาย GA จะมองโครโมโซมเป็นหนึ่งคำตอบประกอบไปด้วยองค์ประกอบที่เรียกว่ายีน (gene) ซึ่งในทางชีววิทยาถือว่าเป็นส่วนที่เก็บคุณลักษณะสำคัญของสิ่งมีชีวิตเอาไว้ เช่น สีของตาหรือผม เป็นต้น ดังนั้นโครโมโซมใน GA จึงเป็นที่เก็บคุณลักษณะของคำตอบของระบบเอาไว้ เพื่อใช้ในการสืบทอดสายพันธุ์กรรมให้ประชากรรุ่นถัดไป กระบวนการภายในของ GA จะมองคำตอบของระบบอยู่ในรูปของโครโมโซมที่เรียกว่า ‘จีโนไทป์’ (genotype) รูปแบบดังกล่าวมีความแตกต่างจากรูปของตัวแปรที่เราสามารถเข้าใจได้ในระบบปกติที่เรียกว่า ‘ฟีโนไทป์’ (phenotype) เช่น ตัวแปรชนิดจำนวนจริงหรือจำนวนเต็มดังแสดงในรูปที่ 3.3 ที่ซึ่งเรามองเห็นแล้วเข้าใจ การเข้ารหัสจึงเป็นการจัดวางรูปแบบคำตอบของระบบให้อยู่ในรูปที่ GA สามารถเข้าใจและทำงานด้วยได้ ตัวอย่างวิธีการเข้ารหัสวิธีหนึ่งก็คือแบบสายอักขระเลขฐานสอง [Holland, 1975] (binary string encoding) ดังแสดงในรูปที่ 3.4 โครโมโซมจะได้อมาจากการเรียงกันด้วยคำตอบของระบบแบบบิตต่อบิต วิธีการเข้ารหัสแบบนี้เป็นวิธีที่ง่ายและเป็นวิธีที่นิยมใช้กันทั่วไปวิธีหนึ่ง วิธีเข้ารหัสอื่น ๆ ก็มีเช่น การเข้ารหัสแบบจำนวนเต็ม [Wright, 1991] (integer) และการเข้ารหัสแบบค่าจริง [Wright, 1991][Janikow and Michalewicz, 1991] (real-valued encoding) ฯลฯ ข้อดีของวิธีการเข้ารหัสทั้งสองแบบ คือ สามารถเข้าใจและตรวจดูค่าได้ทันทีโดยไม่ต้องถอดรหัส วิธีการเข้ารหัสแบบค่าจริงเป็นอีกวิธีที่มีข้อดีเหนือกว่าวิธีอื่นๆ ในแง่ของการคำนวณเชิงตัวเลข อันเนื่องมาจากวิธีดังกล่าวสามารถใช้ตัวแปรแบบจำนวนจริงในการคำนวณโดยตรง จึงไม่มีการสูญเสียความละเอียดของตัวเลขแต่อย่างใด นอกจากนั้นยังสามารถ



รูปที่ 3.5: ฟังก์ชันวัตถุประสงค์กับการเชื่อมโยง GA เข้ากับระบบ

ใช้วิธีคำนวณที่หลากหลายกว่าต่อโครโมโซมได้โดยตรงอีกด้วยวิธีการเข้ารหัสแต่ละวิธีจะมีคุณสมบัติที่ไม่เหมือนกันซึ่งขึ้นอยู่กับปัจจัยหลายๆ อย่าง เช่น ชนิดของคำตอบหรือวิธีการคำนวณในระบบ ฯลฯ โดยปกติแล้วถ้า GA ใช้โครโมโซมที่อยู่ในรูปของฟีโนไทป์ การเข้ารหัสก็เป็นขั้นตอนที่ไม่จำเป็น เนื่องจากโครโมโซมที่เป็นฟีโนไทป์จะอยู่ในรูปเดียวกันกับคำตอบของระบบซึ่งสามารถเข้าใจได้ทันที

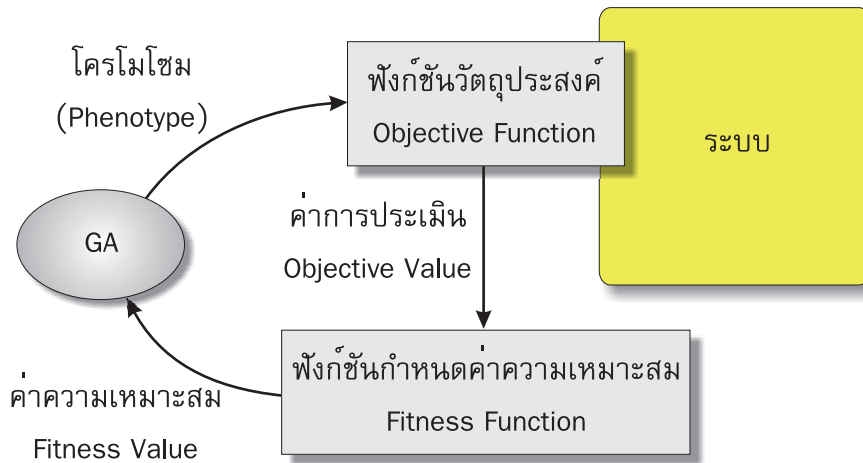
ปกติแล้วใน GA จะใช้โครโมโซมหลายๆ ชุดแทนคำตอบของระบบ คือ $S_i, i = 1, 2, \dots, N$ (N โครโมโซม) นั่นคือในวัฏจักรหนึ่งของ GA จะมีคำตอบอยู่หลายๆ ชุด ปกติขนาดของประชากรจะมีจำนวนอยู่ระหว่าง 30-100 โครโมโซม ในระบบที่เป็นเวลาจริง (real-time system) มีการใช้ไมโคร GA (micro-GA) ซึ่งมีขนาดของประชากรเพียงประมาณ 10 โครโมโซมเพื่อเพิ่มความเร็วในการคำนวณ ขนาดของประชากรอาจจะกล่าวได้ว่าเป็นสิ่งแรกที่ต้องทำการกำหนดก่อน แน่นอนว่าคำตอบของระบบควรจะเป็นคำตอบที่ดีที่สุดจากประชากรที่มีอยู่ ดังนั้นจำนวนของโครโมโซมที่เหมาะสมนั้นจะขึ้นอยู่กับระบบและการออกแบบ จึงไม่มีกฎเกณฑ์ที่แน่นอนในการระบุจำนวนที่เหมาะสมของโครโมโซมในแต่ละครั้งได้ อย่างไรก็ตามก็มีหลักแนวคิดอยู่ว่าจำนวนโครโมโซมที่น้อยเกินไปอาจจะมีผลให้ประชากรที่มีอยู่ทั้งหมดไม่สามารถขยายพันธุ์ครอบคลุมไปถึงคำตอบของระบบได้ตามที่ต้องการ ในขณะที่จำนวนโครโมโซมที่มากเกินไปก็จะก่อให้เกิดความล่าช้าในการคำนวณ และทำให้เกิดความซ้ำซ้อนกันของโครโมโซมซึ่งอาจจะมีผลให้ GA ไม่สามารถลู่เข้าหาคำตอบได้อย่างที่คาดหวัง

3.4 การประเมินค่าความเหมาะสม Fitness Evaluation

การประเมินค่าความเหมาะสมเป็นขั้นตอนในการประเมินว่าโครโมโซมหนึ่งๆ ดีหรือไม่ดีอย่างไร เมื่อเทียบกับโครโมโซมอื่นๆ ที่มีอยู่ในกลุ่มประชากรนั้นๆ โดยปกติแล้วการประเมินค่าความเหมาะสมของโครโมโซมจะประกอบไปด้วยการคำนวณค่าของสองฟังก์ชันดังนี้

3.4.1 ฟังก์ชันวัตถุประสงค์ Objective/Evaluation Function

ฟังก์ชันวัตถุประสงค์เป็นส่วนสำคัญในกระบวนการของ GA ที่ใช้ในการประเมินผลคำตอบของระบบว่าดีหรือไม่ดีแค่ไหน ฟังก์ชันวัตถุประสงค์เป็นฟังก์ชันที่ทำการประเมินคำตอบจากโครโมโซมโดยเทียบกับเป้าหมายของระบบ ในกรณีที่ระบบเป็นปัญหาของการค้นหาค่าน้อยที่สุด (minimization problem) โครโมโซมที่เป็นคำตอบที่ดีที่สุดของระบบจะมีค่าตัวเลขจากฟังก์ชันวัตถุประสงค์ที่น้อยที่สุด ตัวอย่างของฟังก์ชันวัตถุประสงค์ ได้แก่ ฟังก์ชันทั่วไปที่ใช้คำนวณค่าความผิดพลาดของระบบ (error) เช่น MSE (mean-squared error) SSE (sum-squared error) หรือ RMSE (root-mean-squared error) เป็นต้น ดังนั้นวัตถุประสงค์ของระบบดังกล่าวก็คือต้องการให้ค่า



รูปที่ 3.6: ฟังก์ชันวัตถุประสงค์และฟังก์ชันกำหนดค่าความเหมาะสม

ความผิดพลาดของระบบมีค่าน้อยที่สุด ค่าการประเมิน (evaluation value) ที่ได้จากฟังก์ชันวัตถุประสงค์จะเป็นตัวบอกว่าคำตอบของระบบขณะนั้นดีหรือไม่ดีเท่าไร ในกรณีนี้ สำหรับคำตอบหนึ่งๆ แล้ว ค่าความผิดพลาดที่น้อยกว่าหมายถึงคำตอบนั้นจะเป็นคำตอบที่ดีกว่า ค่าการประเมินที่ได้จากฟังก์ชันวัตถุประสงค์บางครั้งจะเรียกว่าค่าวัตถุประสงค์ (objective value) เนื่องจากฟังก์ชันวัตถุประสงค์ต้องใช้ตัวระบบเป็นทั้งคำนวณค่าการประเมิน ดังนั้นฟังก์ชันวัตถุประสงค์จึงถือเป็นส่วนสำคัญในการเชื่อมโยง GA เข้ากับระบบในโลกจริงดังแสดงในรูปที่ 3.5

สังเกตว่าโครโมโซมที่นำไปประเมินค่าด้วยฟังก์ชันวัตถุประสงค์จะต้องอยู่ในรูปที่ระบบเข้าใจ (นั่นคืออยู่ในรูปพีโนไทป์) ดังนั้นถ้ากำหนดให้โครโมโซม S ที่เวลา t ใดๆ คือ $S(t)$ เราสามารถเขียนความสัมพันธ์ของค่าการประเมินของโครโมโซมนี้กับฟังก์ชันวัตถุประสงค์ได้ดังนี้

$$f(S(t)) = f(s_1(t), s_2(t), \dots, s_L(t)) \quad (3.2)$$

โดยที่ $s_1(t), s_2(t), \dots, s_L(t)$ คือคำตอบของระบบที่ผ่านการถอดรหัสให้อยู่ในรูปพีโนไทป์เรียบร้อยแล้ว

3.4.2 ฟังก์ชันกำหนดค่าความเหมาะสม Fitness Function

ฟังก์ชันกำหนดค่าความเหมาะสมเป็นฟังก์ชันที่ทำการจับคู่ค่าการประเมินที่ได้จากฟังก์ชันวัตถุประสงค์ไปเป็นค่าความเหมาะสม (fitness value) จุดประสงค์ของฟังก์ชันนี้ก็คือเพื่อทำการกำหนดค่าความเหมาะสมให้กับโครโมโซมแต่ละตัว โดยทำการเปรียบเทียบกันเองภายในกลุ่มประชากร ค่าความเหมาะสมเหล่านี้จะถูกนำไปใช้เป็นมาตรฐานวัด เพื่อตัดสินใจคัดเลือกโครโมโซมที่จะใช้ในการสืบสายพันธุ์ในรุ่นถัดไป สาเหตุที่ GA ไม่ใช้ค่าการประเมินในการคัดเลือกโครโมโซม ก็เพราะว่าค่าการประเมินที่ได้จากฟังก์ชันวัตถุประสงค์นั้นจะมีค่าขึ้นอยู่กับระบบ จึงทำให้ตัวเลขที่ได้มีความหลากหลายและแตกต่างกันเกินไป ยกตัวอย่างเช่น ขนาดค่าความผิดพลาดของระบบสามารถมีค่าน้อยที่สุดคือศูนย์และมากที่สุดที่ไม่จำกัดขนาดและเครื่องหมาย ดังนั้นเราจะเห็นได้ชัดว่าเป็นการไม่สะดวกนักที่จะนำเอาค่าดังกล่าวมาใช้ในการคัดเลือกโครโมโซม เนื่องจากค่านั้นอาจจะมีความแตกต่างกันเกินไป (โดยเฉพาะในแต่ละรอบของ GA ที่มีการประเมินโครโมโซม ค่าความผิดพลาดอาจจะแตกต่างกันโดยสิ้นเชิงได้) ฟังก์ชันกำหนดค่าความเหมาะสมจึงเป็นการคำนวณค่าการประเมินของโครโมโซมทั้งหมด เทียบกับโครโมโซมด้วยตัวเองและปรับให้มีค่าที่อยู่บนบรรทัดฐานเดียวกันกล่าวคือ

$$E(F_i) = [E_{min}, E_{max}], i = 1, 2, \dots, N \quad (3.3)$$

โดยที่ $E(F_i)$ คือฟังก์ชันกำหนดค่าความเหมาะสมจากการประเมิน F_i (ของโครโมโซมตัวที่ i) E_{min} และ E_{max} เป็นค่าต่ำสุดและค่าสูงสุดของค่าความเหมาะสม โดยปกติจะมีค่าคงที่ตลอดระยะเวลาการทำงานของ GA (ซึ่งต่างกับค่าการประเมินที่สามารถเปลี่ยนแปลงไปตามสภาวะของคำตอบที่คำนวณได้ในแต่ละรอบของ GA) รูปที่ 3.6 แสดงความสัมพันธ์ระหว่างฟังก์ชันวัตถุประสงค์กับฟังก์ชันกำหนดค่าความเหมาะสม

เทคนิคของฟังก์ชันกำหนดค่าความเหมาะสมมีอยู่หลายแบบ เทคนิคที่ง่ายและนิยมใช้ เช่น วิธีกำหนดอย่างเป็นสัดส่วน (proportional) หรือวิธีการกำหนดบรรทัดฐานเชิงเส้น (linear normalization) ฯลฯ รายละเอียดโดยสังเขปของเทคนิคดังกล่าวมีดังต่อไปนี้

▷ **วิธีกำหนดค่าความเหมาะสมอย่างเป็นสัดส่วน - Proportional Fitness Function**

1. กำหนดให้ประชากรประกอบไปด้วยโครโมโซมทั้งหมด N ตัว คือ $S_i, i = 1, 2, \dots, N$
2. กำหนดให้ค่าการประเมินของโครโมโซมตัวที่ i คือ $F_i(S_i)$
3. ค่าความเหมาะสมของแต่ละโครโมโซมสามารถคำนวณได้ดังนี้

$$E(F_i) = \frac{F_i}{\sum_{i=1}^N F_i} \quad (3.4)$$

4. ค่าความเหมาะสมที่ได้จากวิธีนี้จะถูกทำให้อยู่ในบรรทัดฐานเดียวกัน นั่นคือมีค่าไม่เกิน 1 อย่างไรก็ตามวิธีนี้ไม่สามารถใช้กับค่าการประเมินที่ติดลบได้

◁

ตัวอย่างต่อไปแสดงค่าเชิงตัวเลขของการประเมินค่าประชากรอย่างเป็นสัดส่วน

■ **ตัวอย่างที่ 3.1** การกำหนดค่าประเมินประชากรอย่างเป็นสัดส่วน

กำหนดให้ค่าการประเมินของประชากร S คือ $F = \{3, 1, 5, 20, 8\}$ จะได้ว่า

$$\sum_{i=1}^5 F_i = 3 + 1 + 5 + 20 + 8 = 37$$

ดังนั้นค่าความเหมาะสมของแต่ละโครโมโซมคือ

$$E(F_1) = \frac{3}{37} = 0.081$$

$$E(F_2) = \frac{1}{37} = 0.027$$

$$E(F_3) = \frac{5}{37} = 0.135$$

$$E(F_4) = \frac{20}{37} = 0.541$$

$$E(F_5) = \frac{8}{37} = 0.216$$

จากค่าความเหมาะสมที่ได้ เราสามารถสรุปคร่าวๆ ว่าโครโมโซมตัวไหนดีและไม่ดีเท่าไรเมื่อเทียบกับโครโมโซมตัวอื่น ยกตัวอย่างเช่น โครโมโซมตัวที่ 2 และโครโมโซมตัวที่ 4 (จากค่า E_2 และ E_4) ถือเป็นโครโมโซมที่แย่และดีที่สุดในกลุ่มประชากรทั้งหมดตามลำดับ (สำหรับกรณีที่เป็นการค้นหาคำตอบที่มีค่ามากที่สุด)

จากตัวอย่างนี้เราจะเห็นได้ว่า ค่าความเหมาะสมของแต่ละโครโมโซมมีค่าอยู่ในย่านเดียวกัน วิธีการประเมินค่าประชากรอย่างเป็นสัดส่วนถือว่าเป็นวิธีเบื้องต้นที่ไม่ยุ่งยาก ใช้ลดความแตกต่างค่าประเมินของแต่ละประชากรได้

ในระดับหนึ่งเท่านั้น

ค่าประเมินของโครโมโซมที่ได้จากวิธีการกำหนดค่าความเหมาะสมอย่างเป็นสัดส่วน ยังคงมีความแตกต่างกันในระดับหนึ่ง หลายๆ เทคนิควิธีของการกำหนดค่าความเหมาะสม จึงได้ถูกพัฒนาและนำเสนอเพื่อลดข้อจำกัดดังกล่าว พิจารณาวีธีกำหนดค่าความเหมาะสมอย่างเป็นบรรทัดฐานเชิงเส้นต่อไปนี้

▷ **วิธีกำหนดค่าความเหมาะสมอย่างเป็นบรรทัดฐานเชิงเส้น - Linear Normalization Fitness Function**

1. ทำการจัดเรียงโครโมโซมด้วยค่าการประเมินตามลำดับ โดยที่เรียงจากมากไปน้อยถ้าระบบต้องการหาคำตอบที่มีค่ามากที่สุด ในขณะที่เรียงจากน้อยไปมากจะใช้สำหรับระบบที่ต้องการหาคำตอบที่มีค่าน้อยที่สุด กำหนดให้ตัวชี้ที่แสดงลำดับของการเรียงเป็น r
2. ให้โครโมโซมที่ดีที่สุดมีค่าความเหมาะสมเป็น E_{best} แล้วโครโมโซมตัวที่ i จะมีค่าความเหมาะสมคือ

$$E_i = E_{best} - (r - 1) \times \lambda \quad (3.5)$$

โดยที่ λ คือค่าอัตราการลดลง (decrement rate)

3. ค่าความเหมาะสมของโครโมโซมจะมีค่าไม่เกิน E_{best} ในขณะที่โครโมโซมที่อยู่ในอันดับแรกๆ จะมีค่าความเหมาะสมที่ดีกว่าตามสัดส่วนของ $r - 1$
4. ค่าอัตราการลดลงจะมีค่าแตกต่างกันไปตามลักษณะของค่าการประเมิน (ซึ่งอาจจะไม่เหมือนกันในแต่ละระบบ)

ตัวอย่างต่อไปแสดงค่าเชิงตัวเลขของการกำหนดค่าความเหมาะสมอย่างเป็นบรรทัดฐานเชิงเส้น

■ **ตัวอย่างที่ 3.2** การกำหนดค่าประเมินประชากรอย่างเป็นบรรทัดฐานเชิงเส้น

พิจารณาค่าการประเมินของประชากรจากตัวอย่างที่แล้วคือ $F = \{3, 1, 5, 20, 8\}$ ค่าการประเมินที่ดีที่สุดในการประชากรกลุ่มนี้คือ 20 ทำการเรียงลำดับค่าการประเมินได้ดังนี้

$$F_{sorted} = \{F_4, F_5, F_3, F_1, F_2\}$$

ค่าการประเมินที่ดีที่สุดคือค่า 20 อยู่ในลำดับที่ 4 มีค่าตัวชี้เป็น 1 และค่าการประเมินที่แย่ที่สุดคือค่า 1 อยู่ในลำดับที่ 2 มีค่าตัวชี้เป็น 5 กำหนดให้อัตราการลดลงมีค่าเท่ากับ 1 ค่าความเหมาะสมของแต่ละโครโมโซมจะคำนวณได้ดังนี้

$$E_1 = 20 - (4 - 1) \times 1 = 17$$

$$E_2 = 20 - (5 - 1) \times 1 = 16$$

$$E_3 = 20 - (3 - 1) \times 1 = 18$$

$$E_4 = 20 - (1 - 1) \times 1 = 20$$

$$E_5 = 20 - (2 - 1) \times 1 = 19$$

ด้วยวิธีการกำหนดบรรทัดฐานเชิงเส้น จะเห็นว่าสุดท้ายแล้วค่าการประเมินของระบบอยู่ในบรรทัดฐานเดียวกัน โดยมีค่า E_{best} เป็นหลัก ค่าการประเมินอื่นๆ จะถูกจับคู่ไปเป็นค่าความเหมาะสมที่ไม่ขึ้นอยู่กับค่าการประเมิน แต่

จะขึ้นอยู่กับลำดับของค่าการประเมินที่บ่งบอกว่าโครโมโซมนั้นดีหรือไม่ดีอย่างไร ในที่นี้โครโมโซมตัวที่ 4 จะมีความเหมาะสมที่สุด ในขณะที่โครโมโซมตัวที่ 2 จะมีความเหมาะสมที่แย่ที่สุด

เมื่อเปรียบเทียบค่าความเหมาะสมที่ได้ในตัวอย่างนี้ กับค่าที่ได้จากวิธีการกำหนดค่าความเหมาะสมอย่างเป็นสัดส่วน จะเห็นได้อย่างชัดเจนว่า ค่าที่ได้จากวิธีนี้ลดความแตกต่างค่าการประเมินของแต่ละโครโมโซมได้เป็นอย่างดี

ค่าความเหมาะสมของแต่ละโครโมโซมที่ได้ จะถูกนำไปใช้ในหลายๆ ขั้นตอนของ GA ในหัวข้อถัดไป จะได้กล่าวถึงรายละเอียดของการคัดเลือกโครโมโซม ซึ่งใช้ค่าความเหมาะสมที่ได้ในที่นี้เป็นเกณฑ์หลักในการคัดเลือก

3.5 การคัดเลือกสายพันธุ์ Selection

การคัดเลือกสายพันธุ์เป็นขั้นตอนในการคัดเลือกโครโมโซม ที่ดีที่สุดจากภายในกลุ่มประชากรทั้งหมด ซึ่งโครโมโซมที่ได้จะถูกนำไปใช้เป็นตัวต้นกำเนิดสายพันธุ์หรือ ‘พ่อแม่’ เพื่อใช้ในการให้กำเนิดลูกหลานในรุ่นถัดไป โดยปกติแล้วเพื่อให้ได้สายพันธุ์ที่ดี ต้นกำเนิดของสายพันธุ์จะต้องดีด้วย จึงกลายเป็นปัญหาว่าจะทำการคัดเลือกต้นกำเนิดสายพันธุ์ที่ดีได้อย่างไรการคัดเลือกสายพันธุ์เป็นการจำลองการคัดเลือกโครโมโซมที่จะสามารถอยู่รอดได้ในแต่ละรุ่น สำหรับ GA นั้นจะทำการคัดเลือกโครโมโซมโดยการพิจารณาที่ค่าความเหมาะสมของโครโมโซมนั้นๆ ดังนั้นโครโมโซมไหนมีค่าความเหมาะสมที่ดี ย่อมหมายถึงการเป็นโครโมโซมที่ดีและควรมีโอกาสที่จะให้ลูกหลาน (offspring) ในจำนวนที่มากกว่าได้ ซึ่งย่อมเป็นการบ่งบอกว่าโอกาสในการอยู่รอดในรุ่นถัดไปก็จะมีเพิ่มมากขึ้นด้วย ได้มีการนำเสนอเทคนิควิธีการคัดเลือกสายพันธุ์อย่างหลากหลาย ซึ่งทั้งหมดสามารถแบ่งออกได้เป็นสองกลุ่มหลักคือ

- **วิธีที่ใช้ค่าความเหมาะสมโดยตรง** วิธีนี้จะใช้ค่าความเหมาะสม (fitness value) ของแต่ละโครโมโซมสำหรับการคัดเลือกสายพันธุ์
- **วิธีใช้ค่าความเหมาะสมโดยอ้อม** วิธีนี้จะมีการแปลงค่าความเหมาะสมให้อยู่ในช่วงที่ต้องการ เช่นการทำให้เป็นบรรทัดฐานอยู่ในช่วง $[0, 1]$ แล้วใช้สำหรับการคัดเลือกสายพันธุ์ต่อไป

ขั้นตอนในการคัดเลือกสายพันธุ์ประกอบไปด้วย 2 ขั้นตอนใหญ่ๆ ได้แก่ (ดูรูปที่ 3.7 ประกอบ)

1. การกำหนดค่าโอกาส P ในการถูกคัดเลือกเพื่อเป็นตัวต้นกำเนิดสายพันธุ์ให้กับโครโมโซม
2. การแปลงค่าโอกาส P ไปเป็นจำนวนของโครโมโซมลูกหลาน

ในขั้นตอนการคัดเลือกดังกล่าว GA จะใช้ค่าโอกาส P แทนค่าความเหมาะสม ผลลัพธ์จากการคัดเลือกคือจำนวนลูกหลานที่จะได้จากโครโมโซมนั้นๆ โครโมโซมที่มีโอกาสในการถูกคัดเลือกสูงจะให้จำนวนของโครโมโซมลูกหลานที่สูงด้วย ดังนั้นสิ่งที่เราต้องพิจารณาคือการคำนวณหาค่า P ที่เหมาะสม



รูปที่ 3.7: การกำหนดค่าโอกาส P เพื่อใช้ในขั้นตอนการคัดเลือก

วิธีการกำหนดค่าโอกาส P ในการถูกคัดเลือกที่ใช้กันแพร่หลายทั่วไป เช่น [Baker, 1985][Prügel-Bennett, 2000][Prügel-Bennett and Shapiro, 1994][Prügel-Bennett and Shapiro, 1997] วิธีการแบ่งเป็นสัดส่วน (proportionate) วิธีของโบลต์ซมันน์ (Boltzmann) วิธีการจัดอันดับ (ranking) และวิธีจัดการแข่งขัน (tournament) ฯลฯ ส่วนการแปลงค่าโอกาส P ไปเป็นจำนวนโครโมโซมลูกหลาน หรือที่เรียกว่าการชักตัวอย่าง (sampling) จะใช้วิธีของวงล้อรูเล็ต (roulette wheel sampling) หรือวิธีกระบวนการเฟ้นสุ่มครอบคลุมจักรวาล (stochastic universal sampling หรือ SUS)

3.5.1 การกำหนดค่าโอกาสในการถูกคัดเลือก

หลักการโดยทั่วไปของการกำหนดโอกาส P คือการใช้ค่าความเหมาะสมของแต่ละโครโมโซม เพื่อช่วยในการคัดเลือกตัวอย่างวิธีการกำหนดค่าโอกาสมียาวละเอียดต่อไปนี้

- **การคัดเลือกด้วยการแบ่งเป็นสัดส่วน (Proportionate Selection)** วิธีการนี้จะทำการคัดเลือกโครโมโซมอย่างเป็นสัดส่วนจากค่าความเหมาะสมของโครโมโซมนั้นๆ ถ้ากำหนดให้โครโมโซม S มีค่าความเหมาะสมเป็น $E(S)$ ค่าโอกาสในการถูกคัดเลือกของโครโมโซมนี้คือ

$$P(S) = \frac{E(S)}{\bar{E}} \quad (3.6)$$

โดยที่ \bar{E} คือ ค่าความเหมาะสมเฉลี่ยของโครโมโซมทั้งหมด ค่า $P(S)$ แสดงให้เห็นว่าโครโมโซมแต่ละตัวสามารถเป็นต้นกำเนิดสายพันธุ์ในอัตราส่วนที่แตกต่างกัน โครโมโซมที่มีค่าความเหมาะสมมากก็จะมีโอกาสในการสืบสายพันธุ์ด้วยอัตราที่สูงกว่าโครโมโซมที่มีค่าความเหมาะสมน้อยกว่าข้อจำกัดของการคัดเลือกสายพันธุ์ด้วยวิธีการนี้ คือ ค่าความเหมาะสมจะต้องมีค่าเป็นบวกเท่านั้น อย่างไรก็ตามเราสามารถใส่ค่าเอกซ์โพเนนเชียลของค่าความเหมาะสมซึ่งจะมีค่าเป็นบวกเสมอแทนได้ (วิธีดังกล่าวจะกลายเป็นวิธีของโบลต์ซมันน์นั่นเอง) วิธีการแบ่งเป็นสัดส่วนเป็นวิธีที่ง่ายแต่อาจจะสามารถนำไปสู่คำตอบแบบวงแคบเฉพาะถิ่นได้ (local optimum) วิธีการแบบอื่นๆ จึงถูกพัฒนาและศึกษาในประสิทธิภาพต่อการทำงานของ GA ดังเช่นวิธีของโบลต์ซมันน์หรือวิธีแบบจัดอันดับ

- **การคัดเลือกแบบโบลต์ซมันน์ (Boltzmann Selection)** วิธีของโบลต์ซมันน์เป็นวิธีการแก้ปัญหาของโครโมโซมที่มีค่าความเหมาะสมที่เป็นลบ นอกจากนั้นแล้วยังมีจุดประสงค์เพื่อลดความแตกต่างของค่าความเหมาะสมของประชากรโดยรวม พิจารณาค่าโอกาสในการถูกคัดเลือกของโครโมโซม S ที่มีค่าความเหมาะสมเป็น $E(S)$ สามารถเขียนได้ดังนี้

$$P(S) = \frac{e^{E(S)}}{\bar{E}} \quad (3.7)$$

สมการข้างต้นได้มาจากการคำนวณค่าเอกซ์โพเนนเชียลของค่าความเหมาะสมนั่นเอง

- **การคัดเลือกแบบจัดอันดับ (Ranking Selection)** วิธีการนี้เป็นวิธีการที่ค่อนข้างง่าย โดยโครโมโซมจะถูกจัดเรียงให้มีย่ออันดับ r ตามค่าความเหมาะสม โครโมโซมที่มีค่าความเหมาะสมที่ดีที่สุดจะมีอันดับ N โดยที่ N คือจำนวนโครโมโซมทั้งหมด (N จะเป็นค่าอันดับที่มากที่สุด) ในขณะที่โครโมโซมที่มีค่าความเหมาะสมที่ด้อยที่สุดจะมีอันดับ 1 โอกาสในการถูกคัดเลือกของโครโมโซมจะมีค่าเท่ากับ

$$P(S) = \frac{r}{\bar{E}} \quad (3.8)$$

วิธีการจัดอันดับมีข้อดีที่ค่า P จะไม่แปรผันกับขนาดของค่าความเหมาะสมแต่จะขึ้นกับอันดับของโครโมโซม อย่างไรก็ตามวิธีการดังกล่าวจะมีผลทำให้การเข้าสู่คำตอบของ GA ช้า เนื่องจากโครโมโซมที่ด้อยกว่ามีโอกาสในการถูกคัดเลือกที่ดีขึ้นเมื่อเทียบกับวิธีที่กล่าวมาก่อนข้างต้น

- **การคัดเลือกแบบจัดการแข่งขัน (Tournament Selection)** เป็นวิธีการเดียวกับการแข่งขันกีฬาทั่วไป ทำได้โดยการสุ่มแบ่งกลุ่มคัดเลือกโครโมโซม แล้วเลือกเอาโครโมโซมที่ดีที่สุดในกลุ่มนั้นเพื่อหาโครโมโซมผู้ชนะเป็นต้นกำเนิดสายพันธุ์ต่อไป จำนวนของโครโมโซมในแต่ละกลุ่มนั้นจะแตกต่างกันออกไป โดยปกติแล้วจะใช้วิธีสุ่มแบบจับคู่โครโมโซม (นั่นคือมีเพียง 2 โครโมโซมที่ถูกสุ่มเลือกเข้ามาในแต่ละการแข่งขัน) หลักการจัดการแข่งขันมีดังต่อไปนี้

- ทำการสุ่มเลือกโครโมโซม K ตัวสำหรับจัดการแข่งขันขนาด K (tournament size)
- เลือกโครโมโซมที่มีค่าประเมินดีที่สุดจากการแข่งขัน ด้วยค่าความน่าจะเป็นเท่ากับ p
- เลือกโครโมโซมที่มีค่าประเมินดีเป็นอันดับสอง (รองอันดับหนึ่ง) ด้วยค่าความน่าจะเป็นเท่ากับ $p \times (1 - p)$
- เลือกโครโมโซมที่มีค่าประเมินดีเป็นอันดับถัดไป (รองอันดับสอง) ด้วยค่าความน่าจะเป็นเท่ากับ $p \times (1 - p)^2$
- เลือกโครโมโซมที่มีค่าประเมินดีเป็นรองอันดับที่ n ด้วยค่าความน่าจะเป็นเท่ากับ $p \times (1 - p)^n$

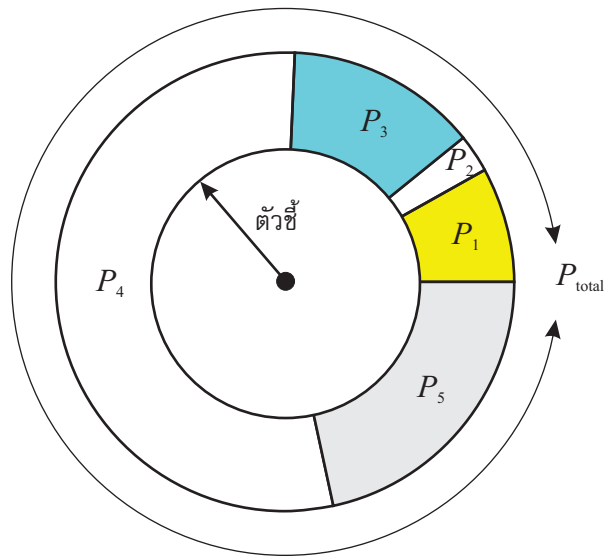
ถ้ากำหนดให้ค่า $p = 1$ การคัดเลือกจะเลือกโครโมโซมที่ดีที่สุดเสมอ ในขณะที่ถ้า $K = 1$ การคัดเลือกเปรียบเสมือนกับการสุ่มเลือกโครโมโซมนั่นเอง โครโมโซมที่ชนะการแข่งขันและถูกคัดเลือกแล้ว สามารถออกจากการแข่งขันหรืออยู่ต่อ (เพื่อสามารถถูกเลือกได้อีก) ก็ได้ เราสามารถปรับขนาดของการแข่งขัน K เพื่อให้ได้ความเข้มข้นในการแข่งขันที่เหมาะสม กล่าวคือถ้า K มีขนาดใหญ่หรือการแข่งขันขนาดใหญ่ โครโมโซมที่ไม่ดีมาก (ค่าการประเมินน้อย) ก็จะมีโอกาสชนะน้อย

วิธีการจัดการแข่งขันมีความเหมาะสมในการทำให้ปัญหาความเหลื่อมล้ำของค่าความเหมาะสมของโครโมโซมหมดไป นอกไปจากนั้นแล้ว วิธีการแข่งขันยังเขียนโปรแกรมได้ง่ายและทำงานเป็นแบบขนานได้ด้วย

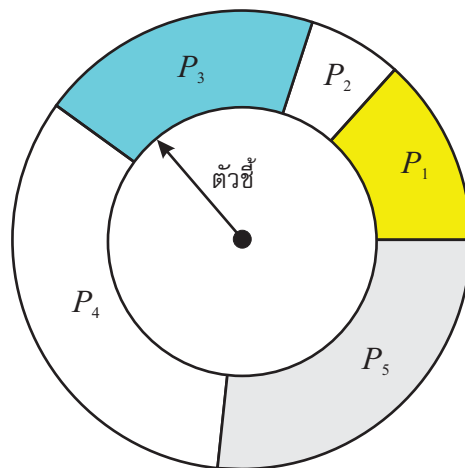
3.5.2 การแปลงค่าโอกาสเป็นจำนวนโครโมโซมลูกหลาน

หลังจากที่ได้ทำการกำหนดค่าโอกาส P ในการถูกคัดเลือกให้กับแต่ละโครโมโซมจนหมดแล้ว ขั้นตอนต่อไปก็คือการชักตัวอย่าง ซึ่งเป็นการนำเอาค่าโอกาสนั้นไปทำการแปลงให้เป็นค่าตัวเลข ตัวเลขดังกล่าวจะแสดงถึงจำนวนของลูกหลานที่โครโมโซมนั้นๆ จะสามารถให้กำเนิดในขั้นต่อไปได้ วิธีการแปลงค่าโอกาสให้เป็นจำนวนโครโมโซมลูกหลานที่นิยมใช้มีดังต่อไปนี้

- **วิธีการชักตัวอย่างแบบวงล้อรูเล็ต (Roulette Wheel Sampling)** ในขั้นตอนแรกจะทำการสร้างวงล้อรูเล็ตขึ้นมาก่อน โดยกำหนดให้ P_{total} คือผลรวมของค่าโอกาสในการถูกคัดเลือกของโครโมโซมในประชากรทั้งกลุ่ม ค่านี้จะมีค่าเทียบเท่ากับเส้นรอบวงของวงล้อรูเล็ต หลังจากนั้นค่า P ของโครโมโซมแต่ละตัวจะถูกแปลงไปยังบนวงล้อรูเล็ตภายในช่วง $[0, P_{total}]$ โดยที่ขนาดบนวงล้อรูเล็ตสำหรับแต่ละโครโมโซมจะสัมพันธ์กับค่า P ของโครโมโซมนั้นๆ รูปที่ 3.8 แสดงตัวอย่างของวงล้อรูเล็ตสำหรับกลุ่มประชากร S ที่มีค่าความเหมาะสม $E = \{3, 1, 5, 20, 8\}$ ค่า P_i คือค่าโอกาสในการถูกคัดเลือกของโครโมโซมตัวที่ i ซึ่งได้มาจากวิธีการแบ่งเป็นสัดส่วน สังเกตว่าค่า P ของโครโมโซมจะสัมพันธ์โดยตรงกับค่าความเหมาะสมของโครโมโซม จากรูปดังกล่าว โครโมโซม S_4 ซึ่งมีค่าความเหมาะสมเท่ากับ 20 เป็นค่าสูงที่สุดในกลุ่มจะมีขนาดบนวงล้อรูเล็ตมากที่สุดและมีโอกาสในการถูกคัดเลือกสูงที่สุด (ด้วยค่า P_4) ในขณะที่โครโมโซม S_2 ซึ่งมีค่าความเหมาะสมเท่ากับ 1 เป็นค่าต่ำที่สุดในกลุ่มจะมีขนาดบนวงล้อรูเล็ตเล็กที่สุดและมีโอกาสในการถูกคัดเลือกต่ำที่สุด (ด้วยค่า P_2) ขั้นตอนในการคัดเลือกโครโมโซมจะเริ่มจากการสุ่มค่าตัวชี้ซึ่งเป็นตัวเลขระหว่าง 0 ถึง P_{total} และถ้าตัวเลขดังกล่าวตรงกับโครโมโซมใดบนวงล้อรูเล็ต โครโมโซมนั้นจะถูกเลือก กระบวนการนี้เปรียบได้กับการหมุนวงล้อในเกมรูเล็ตนั่นเอง

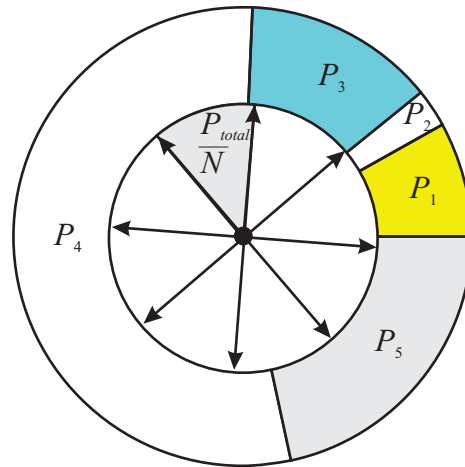


รูปที่ 3.8: วงล้อรูเล็ตจากค่าความเหมาะสมของแต่ละโครโมโซม



รูปที่ 3.9: วงล้อรูเล็ตจากวิธีการจัดอันดับ

ในแต่ละครั้งของการหมุนวงล้อก็จะได้โครโมโซมที่จะเป็นต้นกำเนิดสายพันธุ์มาหนึ่งตัว การสุ่มตัวเลขเพื่อคัดเลือกโครโมโซมจะดำเนินไปเรื่อยๆ จนกระทั่งได้ต้นกำเนิดสายพันธุ์เท่ากับจำนวนที่ต้องการ ดังนั้นสำหรับการคัดเลือกต้นกำเนิดสายพันธุ์จำนวน N โครโมโซมจะต้องทำการหมุนวงล้อรูเล็ตทั้งหมด N ครั้ง เราจะเห็นได้ชัดเจนว่าโครโมโซมที่มีค่าความเหมาะสมที่สูง จะมีโอกาสถูกคัดเลือกมากกว่าโครโมโซมที่มีค่าความเหมาะสมที่ต่ำกว่า ซึ่งเป็นปรากฏการณ์ปกติในธรรมชาติทั่วไปค่า P ที่ใช้จากตัวอย่างในรูปที่ 3.8 ได้มาจากการแบ่งเป็นสัดส่วน เราจะเห็นได้ชัดเจนว่าโอกาสของโครโมโซมตัวที่สองที่จะถูกเลือกนั้น อาจมีค่าน้อยมากอันจะทำให้เกิดความเหลื่อมล้ำในการคัดเลือกขึ้น วิธีอื่นๆ ที่มีประสิทธิภาพดีกว่าจึงเป็นที่นิยมใช้มากกว่า ยกตัวอย่างเช่นวิธีการจัดอันดับ ซึ่งจะให้ค่าของโอกาสในการถูกคัดเลือกที่ไม่แปรผันไปตามขนาดของค่าความเหมาะสม ดังแสดงในรูปที่ 3.9 เราจะเห็นได้ว่าโอกาสการถูกคัดเลือกของ P_2 เพิ่มขึ้นเป็นสัดส่วนที่เหมาะสม (ถูกจัดเป็นอันดับสุดท้าย) ในขณะที่ P_4 และ P_5 มีค่าโอกาสในการถูกคัดเลือกที่ใกล้เคียงกัน อันเนื่องมาจากมีอันดับที่ติดกัน วิธีวงล้อรูเล็ตเป็นขบวนการสุ่มคัดเลือกโครโมโซม ในบางโอกาสอาจจะมีโครโมโซมใดโครโมโซมหนึ่งเท่านั้นที่บังเอิญถูกสุ่มในการคัดเลือกทุกครั้ง ถ้าเกิดกรณีดังกล่าวขึ้น ประชากร



รูปที่ 3.10: วงล้อสุ่มเลือกจากกระบวนการเฟ้นสุ่มครอบจักรวาล

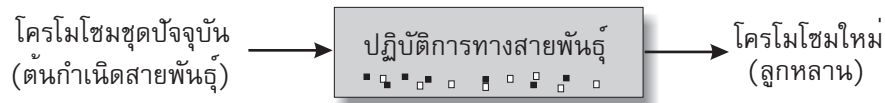
ในรุ่นถัดไปจะประกอบไปด้วยโครโมโซมชนิดเดียวกันหมดทั้งกลุ่ม ซึ่งไม่เป็นประโยชน์ต่อการค้นหาคำตอบแต่อย่างใด วิธีวงล้อสุ่มเลือกสามารถปรับปรุงได้หลายวิธี วิธีหนึ่งที่ยง่ายและมีประสิทธิภาพคือ กำหนดให้ในแต่ละครั้งที่โครโมโซมถูกคัดเลือก ขนาดของโครโมโซมนั้นๆ บนวงล้อสุ่มเลือกจะมีค่าลดลงจนกระทั่งมีขนาดเป็นศูนย์ วิธีนี้ทำให้แต่ละโครโมโซมที่เหลืออยู่ มีขอบเขตในการถูกเลือกที่เท่าเทียมกัน ในขณะเดียวกัน ก็ช่วยลดโอกาสความเป็นใหญ่ในหมู่ประชากรของโครโมโซมตัวใดตัวหนึ่งได้

- **วิธีการชักตัวอย่างแบบกระบวนการเฟ้นสุ่มครอบจักรวาล (Stochastic Universal Sampling หรือ SUS)**
วิธี SUS เป็นการชักตัวอย่างที่ถูกเรียกใช้เพียงครั้งเดียว ก็สามารถคัดเลือกโครโมโซมตามจำนวนที่ต้องการได้ วิธีชักตัวอย่างแบบ SUS มีข้อแตกต่างไปจากแบบวงล้อสุ่มเลือกตรงที่มีการใช้ตัวชี้มากกว่าหนึ่งตัว นั่นคือสำหรับการคัดเลือก N โครโมโซมจะมีตัวชี้ทั้งหมด N ตัว โดยที่ตัวชี้แต่ละตัวจะมีระยะห่างเท่ากันและมีค่าเท่ากับ $\frac{P_{total}}{N}$ ดังแสดงในรูปที่ 3.10 SUS จะเริ่มจากการสุ่มสลับที่โครโมโซมบนวงล้อสุ่มเลือก ถัดมาตัวชี้ ptr ซึ่งเป็นตัวชี้เริ่มต้นจะถูกสุ่มขึ้นมาในช่วง $[0, P_{total}]$ ตัวชี้ที่เหลือจำนวน $N - 1$ ตัวถัดจากตัวชี้ ptr จะถูกคำนวณโดยมีระยะห่างระหว่างตัวชี้ที่ติดกันเท่ากับ $\frac{P_{total}}{N}$ โครโมโซมที่จะถูกคัดเลือกได้แก่โครโมโซมจำนวนทั้งสิ้น N โครโมโซมซึ่งถูกตัวชี้ N ตัวชี้อยู่ วิธีนี้ทำให้ลดความเหลื่อมล้ำในแต่ละโครโมโซมสำหรับการคัดเลือกลงเมื่อเทียบกับวิธีดั้งเดิมของวงล้อสุ่มเลือก โครโมโซมที่มีค่า P สูงจะถูกชี้ด้วยจำนวนตัวชี้ในจำนวนที่มากกว่าโครโมโซมที่มีค่า P ที่ต่ำกว่า

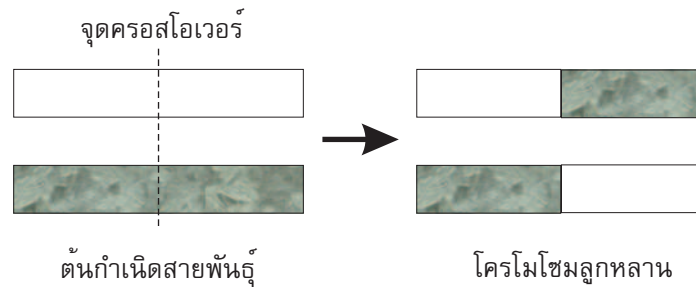
หลังจากขั้นตอนของการคัดเลือกแล้ว โครโมโซมที่ถูกคัดเลือกจะกลายเป็นต้นกำเนิดสายพันธุ์ ซึ่งมีหน้าที่ในการให้กำเนิดโครโมโซมลูกหลานด้วยกรรมวิธีเฉพาะของ GA ดังรายละเอียดในหัวข้อถัดไป

3.6 ปฏิบัติการทางสายพันธุ์ Genetic Operation

หลังจากขบวนการคัดเลือกได้ดำเนินไปจนเสร็จสมบูรณ์ โครโมโซมลูกหลานจะถูกสร้างขึ้นใหม่ จากโครโมโซมที่ถูกคัดเลือกมาเป็นต้นกำเนิดสายพันธุ์ โดยการนำเอาโครโมโซมที่เป็นต้นกำเนิดสายพันธุ์นั้นมาทำการเปลี่ยนแปลงให้เกิดโครโมโซมใหม่ขึ้นมา กลายเป็นโครโมโซมลูกหลานดังแสดงในรูปที่ 3.11 ขั้นตอนดังกล่าวนี้เป็นขั้นตอนสำคัญอีกขั้นตอนหนึ่งในวัฏจักรของ GA ที่ซึ่งมีการคาดหวังว่าโครโมโซมลูกหลานที่เกิดขึ้นมานั้น จะได้รับส่วนดีของโครโมโซมต้นกำเนิดสายพันธุ์โดยผ่านปฏิบัติการทางสายพันธุ์นี้ ถ้าพิจารณาถึงการเปลี่ยนแปลงอันเนื่องมาจาก



รูปที่ 3.11: ปฏิบัติการทางสายพันธุ์



รูปที่ 3.12: ครอสโอเวอร์แบบจุดเดียว

ปฏิบัติการทางสายพันธุ์ที่เกิดขึ้นกับประชากร ซึ่งเป็นคำตอบของระบบแล้ว เราสามารถเปรียบเทียบปฏิบัติการทางสายพันธุ์ได้กับการก้าวเดินไปสู่คำตอบของระบบนั่นเอง โดยปกติทั่วไปปฏิบัติการทางสายพันธุ์ของ GA จะมีอยู่ 2 วิธีหลักๆ คือ การทำครอสโอเวอร์และการทำมิวเทชัน ดังรายละเอียดต่อไปนี้

3.6.1 ครอสโอเวอร์ Crossover

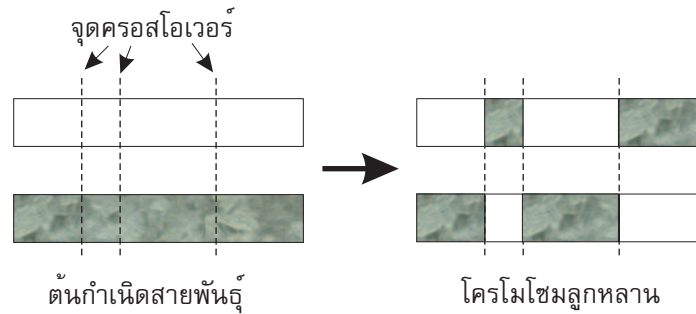
ครอสโอเวอร์เป็นวิธีการรวมตัวใหม่ของโครโมโซม (recombination operator) โดยทำการรวมส่วนย่อยระหว่างโครโมโซมต้นกำเนิดสายพันธุ์ ตั้งแต่สองโครโมโซมขึ้นไป เพื่อให้กลายเป็นโครโมโซมลูกหลาน โครโมโซมลูกหลานที่ได้จากการครอสโอเวอร์นี้จะมีพันธุกรรมจากต้นกำเนิดสายพันธุ์อยู่ในตัว โดยปกติทั่วไปแล้วจะมีการกำหนดอัตราการทำครอสโอเวอร์เอาไว้ซึ่งส่วนใหญ่จะใช้ความน่าจะเป็น (P_c) เป็นตัวกำหนดอัตราดังกล่าว วิธีการทำครอสโอเวอร์มีได้หลายแบบดังรายละเอียดในตัวอย่างต่อไปนี้

- การทำครอสโอเวอร์แบบจุดเดียว (Single-Point Crossover)

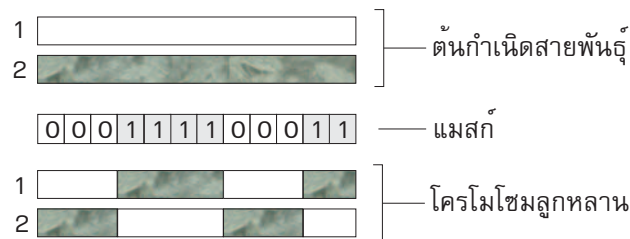
การทำครอสโอเวอร์แบบจุดเดียวนั้น โครโมโซมลูกหลานจะมีสายพันธุ์ของแต่ละต้นกำเนิดอยู่อย่างละหนึ่งส่วน จุดตัดในการทำครอสโอเวอร์นั้นโดยปกติจะได้มาจากการสุ่มเลือก ตัวอย่างของการทำครอสโอเวอร์แบบจุดเดียวแสดงอยู่ในรูปที่ 3.12

- การทำครอสโอเวอร์แบบหลายจุด (Multiple-Point Crossover)

พิจารณาตัวอย่างการทำครอสโอเวอร์แบบหลายจุด ดังแสดงในรูปที่ 3.13 ที่ซึ่งมีการใช้จุดตัดทั้งหมด 3 จุด ดังนั้นโครโมโซมลูกหลานจะมีสายพันธุ์ของต้นกำเนิดอยู่มากกว่าหนึ่งส่วน หลักการเลือกจุดของครอสโอเวอร์นั้นมีอยู่หลายแบบ แต่ละแบบจะให้ผลต่อการเปลี่ยนแปลงของสายพันธุ์ในโครโมโซมลูกหลานที่แตกต่างกันออกไปด้วย วิธีที่ง่ายและเป็นที่ยอมรับทั่วไปคือการสุ่มเลือกจุดครอสโอเวอร์ การทำครอสโอเวอร์แบบหลายจุดจะให้ผลของลูกหลานที่มีความหลากหลายกว่าการทำครอสโอเวอร์แบบจุดเดียว อันจะมีผลให้การเข้าสู่คำตอบของระบบสามารถครอบคลุมพื้นที่ของคำตอบได้มากยิ่งขึ้น อย่างไรก็ตามการทำครอสโอเวอร์แบบหลายจุด ซึ่งทำให้เกิดการเปลี่ยนแปลงของโครโมโซมลูกหลานได้มากกว่าการทำครอสโอเวอร์แบบจุดเดียวนั้น อาจจะทำให้มีโอกาสเบี่ยงเบนของคำตอบที่มีอยู่ในโครโมโซมลูกหลานได้ในอัตราที่สูงกว่าเช่นกัน



รูปที่ 3.13: ครอสโอเวอร์แบบหลายจุด



รูปที่ 3.14: ครอสโอเวอร์แบบสม่ำเสมอ

• ครอสโอเวอร์แบบสม่ำเสมอ [Hussein et al., 2001] (Uniform Crossover)

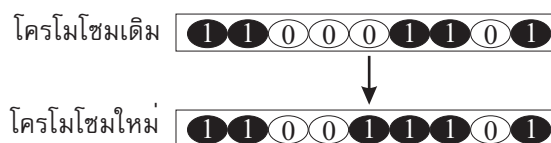
การทำครอสโอเวอร์แบบจุดเดียวและหลายจุด มีการกำหนดจุดตัดเอาไว้ก่อนที่จะทำการสลับส่วนย่อยของโครโมโซมที่จุดนั้นๆ วิธีดังกล่าวมีความแตกต่างไปจากการทำครอสโอเวอร์แบบสม่ำเสมอ ที่ซึ่งถูกออกแบบให้ทุกจุดบนโครโมโซมสามารถเป็นจุดตัดได้ ในทางปฏิบัติจะมีการใช้ครอสโอเวอร์แมสก์หรือตัวพราง (crossover mask) ช่วยในการทำครอสโอเวอร์ ตัวพรางดังกล่าวจะเป็นชนิดไบนารีและมีขนาดจำนวนบิตเท่ากับความยาวของโครโมโซม ค่าของตัวพรางที่ตำแหน่งต่างๆ จะเป็นตัวบอกถึงการครอสโอเวอร์ระหว่างต้นกำเนิดสายพันธุ์ ดังตัวอย่างในรูปที่ 3.14 ณ ตำแหน่งที่ตัวพรางมีค่าเป็น 1 โครโมโซมลูกหลานจะได้รับการสลับส่วนย่อยของโครโมโซมต้นกำเนิดสายพันธุ์ ถ้าตำแหน่งที่แมสก์มีค่าเป็น 0 โครโมโซมลูกหลานจะยังคงเป็นส่วนย่อยของโครโมโซมต้นกำเนิดสายพันธุ์ โดยไม่มีการสลับส่วนแต่อย่างใด

การทำครอสโอเวอร์เป็นผลจากปฏิบัติการทางสายพันธุ์ที่ใช้โครโมโซมต้นกำเนิดสายพันธุ์มากกว่า 1 โครโมโซมขึ้นไป ปฏิบัติการทางสายพันธุ์ในหัวข้อต่อไปจะได้กล่าวถึงรายละเอียดการทำมิวเทชัน ที่ซึ่งเกิดขึ้นกับโครโมโซมตัวเดียวเท่านั้น

3.6.2 มิวเทชัน Mutation

มิวเทชันเป็นวิธีการแปรผันยีนหรือส่วนย่อยของโครโมโซม ซึ่งสามารถเปรียบเทียบได้กับการกลายพันธุ์ของสิ่งมีชีวิตในทางชีววิทยานั้นเอง ปกติแล้วอัตราการทำมิวเทชันจะมีค่าค่อนข้างต่ำ หรืออาจจะกล่าวได้ว่าความน่าจะเป็นในการทำมิวเทชันนั้นมีค่าน้อย ถ้ามิวเทชันคือการเปลี่ยนแปลงยีนในโครโมโซมแล้ว มิวเทชันจะเป็นการเปลี่ยนแปลงเชิงตัวเลขของโครโมโซมนั้นเอง เพราะในทางปฏิบัติแล้ว ยีนของโครโมโซมก็คือบิตในระบบตัวเลขของคอมพิวเตอร์ ดังแสดงในรูปที่ 3.15

การทำมิวเทชันเปรียบเสมือนกับการก้าวเดินไปสู่คำตอบของระบบเช่นเดียวกับการทำครอสโอเวอร์ นอกเหนือไปจากนั้นแล้ว มิวเทชันยังสามารถถูกพิจารณาเป็นการทำให้เกิดความหลากหลายขึ้นในกลุ่มประชากร มีผลให้คำตอบที่เกิดขึ้นในขบวนการของ GA ครอบคลุมพื้นที่การค้นหาคำตอบทั่วถึงยิ่งขึ้น อย่างไรก็ตามอัตราในการทำ



รูปที่ 3.15: การแปรผันยีนในมิวเทชัน

มิวเทชันเป็นปัจจัยที่สำคัญอีกอย่างหนึ่งที่ต้องคำนึงถึง เพราะจะมีผลต่อพฤติกรรมการทำงานของ GA มีผลการค้นคว้ารายงานว่า อัตราการทำมิวเทชันจะขึ้นอยู่กับขนาดของประชากร เพื่อให้การสำรวจพื้นที่ในการค้นหาคำตอบเป็นไปอย่างทั่วถึง [Prügel-Bennett, 2001] ดังนั้นการกำหนดอัตราการทำมิวเทชันต้องมีความเหมาะสมที่สุดต่อระบบด้วย เพื่อก่อให้เกิดผลในการค้นหาคำตอบที่มีประสิทธิภาพยิ่งขึ้น จากที่กล่าวมาทั้งหมด สามารถสรุปได้ว่าจุดประสงค์หลักๆ ของการทำมิวเทชันก็คือเพื่อให้ GA สามารถหลุดพ้นออกจากคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่นได้ (local optimum) โดยการป้องกันไม่ให้โครโมโซมประชากรเกิดการเปลี่ยนแปลง ในทิศทางที่มีความคล้ายคลึงกันไปหมด ซึ่ง ณ จุดนั้น เราอาจจะกล่าวได้ว่าวิวัฒนาการของคำตอบที่ต้องได้หยุดลง

ครอสโอเวอร์และมิวเทชันเป็นการสร้างและการเปลี่ยนแปลงของต้นกำเนิดสายพันธุ์ ซึ่งให้ผลเป็นโครโมโซมลูกหลาน การทำครอสโอเวอร์มีผลให้โครโมโซมลูกหลานได้รับสายพันธุ์จากต้นกำเนิด โดยโครโมโซมลูกหลานจะได้รับส่วนที่ดีจากส่วนย่อยของต้นกำเนิดสายพันธุ์ไป ในขณะที่การทำมิวเทชันเป็นการสร้างความแปรผันขึ้นในโครโมโซมลูกหลาน เพื่อให้เกิดประชากรใหม่ที่ดีกว่าขึ้น ปฏิบัติการทางสายพันธุ์ทั้งสองถูกใช้ใน GA โดยหวังว่าการเปลี่ยนแปลงที่เกิดขึ้น จะมีผลทำให้โครโมโซมลูกหลานมีสายพันธุ์ที่ดีขึ้น อันจะนำไปสู่คำตอบที่ดีที่สุดต่อไป

3.7 การแทนที่ Replacement

การแทนที่เป็นขั้นตอนหลังจากที่ GA ได้โครโมโซมลูกหลานเรียบร้อยแล้ว และจะนำโครโมโซมลูกหลานใหม่ไปแทนที่ประชากรรุ่นเก่า จุดประสงค์ในการแทนที่นั้นค่อนข้างชัดเจน กล่าวคือการนำโครโมโซมลูกหลานมาแทนที่ประชากรรุ่นก่อน จะทำให้ประชากรรุ่นใหม่ประกอบไปด้วยโครโมโซมใหม่ๆ ซึ่งเป็นโครโมโซมที่ดีกว่าเพราะได้สืบสายพันธุ์ที่ดีจากต้นกำเนิดสายพันธุ์ที่ผ่านการคัดเลือกแล้ว กลยุทธ์ในการคัดเลือกว่าโครโมโซมไหนจะถูกแทนที่นั้นสามารถแบ่งได้เป็น 2 วิธี คือ

- การแทนที่ประชากรทั้งรุ่น (generational GA)

เป็นการนำประชากรลูกหลานไปแทนที่ประชากรเก่าทั้งหมด ดังนั้นถ้าในระบบหนึ่งมีจำนวนประชากรเท่ากับ N จำนวนของโครโมโซมลูกหลานที่จะมาแทนที่ก็ต้องมีขนาด N เช่นกัน วิธีนี้เป็นวิธีที่ง่ายเนื่องมาจากไม่จำเป็นต้องมีขั้นตอนของการคัดเลือก ว่าประชากรส่วนไหนจะถูกแทนที่ แต่การที่ไม่มีขั้นตอนดังกล่าวกลายเป็นข้อเสีย นั่นคือ โครโมโซมที่ดีๆ ในรุ่นก่อนจะถูกแทนที่ไปด้วย วิธีนี้ค่อนข้างง่าย คือ ก่อนที่จะทำการแทนที่ให้คัดเลือกเก็บโครโมโซมที่ดีที่สุด 2-3 ตัวแรกเอาไว้โดยอาจจะใช้วิธีการคัดเลือกดังที่อธิบายมาแล้วก่อนหน้านี้ วิธีดังกล่าวอาจเรียกได้ว่าเป็นกลยุทธ์คัดเลือกหัวกระติ (elitist strategy) อย่างไรก็ตามประชากรที่เหลืออยู่อาจจะถูกครอบงำด้วยโครโมโซมหัวกระตินี้ได้โดยง่าย กล่าวคือถ้าไม่มีโครโมโซมใหม่ที่ดีกว่าเกิดขึ้น โครโมโซมที่ดีที่สุดจากรุ่นก่อนก็จะถูกเก็บไว้อยู่ตลอดไป และไม่ก่อให้เกิดการเปลี่ยนแปลงใดๆ ขึ้น ทำให้ GA ไม่สามารถสร้างวิวัฒนาการโครโมโซมใหม่ๆ ขึ้นมาได้ ถึงแม้ว่าผลของโครโมโซมหัวกระติจะมีโอกาสเกิดขึ้นได้ แต่วิธีนี้ก็ได้รับการพิสูจน์แล้วว่าทำให้ระบบโดยรวมดีขึ้น

- การแทนที่ประชากรแบบบางส่วน (partial GA)

เป็นการนำเอาประชากรลูกหลานไปแทนที่ประชากรเดิมเพียงบางส่วนเท่านั้น ดังนั้นจะต้องมีการคัดเลือก

ประชากรที่จะถูกแทนที่ ซึ่งโดยปกติจะพิจารณาจากค่าความเหมาะสมของโครโมโซมนั้นเอง โครโมโซมเก่าจะถูกแทนที่ด้วยโครโมโซมใหม่เพียง 1 หรือ 2 ตัวเท่านั้น กลวิธีในการแทนที่มีอยู่หลายวิธี [Rogers and Prügel-Bennett, 1999][Rogers and Prügel-Bennett, 2000] เช่นการแทนที่ประชากรที่ด้อยที่สุดหรือการแทนที่ประชากรโดยการสุ่มเลือก เป็นต้น

เราได้พิจารณารายละเอียดองค์ประกอบต่างๆ ของ GA เรียบร้อยแล้ว เนื้อหาในส่วนต่อไปจะเป็นตัวอย่างการนำเอา GA ไปประยุกต์ใช้งานด้านต่างๆ ในการใช้ GA กับแต่ละปัญหานั้น สิ่งที่จะต้องพิจารณาก็คือพารามิเตอร์ของ GA ดังที่ได้กล่าวมาแล้วทั้งหมด การเลือกพารามิเตอร์ของ GA สำหรับแต่ละปัญหานั้นจะมีความเหมาะสมแตกต่างกันไป ผู้ใช้งาน GA จึงควรต้องเข้าใจธรรมชาติการทำงานของ GA ให้ดี แล้วทำการปรับแต่งพารามิเตอร์ต่างๆ เพื่อให้ได้ผลลัพธ์ที่เหมาะสมที่สุดให้ได้

3.8 ซอฟต์แวร์ในการช่วยคำนวณ GA

เนื่องมาจากการใช้ GA ทำให้ไม่จำเป็นจะต้องมีผลเฉลยรูปแบบปิดของระบบ ดังนั้นงานที่นำเอา GA มาใช้ส่วนใหญ่จึงเป็นการจำลองสถานการณ์ และถึงแม้ว่าระบบจะมีความแตกต่างกัน แต่ในส่วนของ GA แล้วจะยังคงมีความเหมือนกันอยู่ ดังนั้นจึงมีซอฟต์แวร์ที่ช่วยในการคำนวณ GA อยู่มากมายหลายแบบ เช่น GENESIS [Grefenstette, 1990] GENESys [Thomas, 1992] BUGS [Smith, 1991] TOLKIEN [Tang, 1994] เป็นต้น แต่ละซอฟต์แวร์มีความสามารถและจุดประสงค์การใช้งานที่ต่างกันออกไป สำหรับผู้อ่านที่คุ้นเคยกับการใช้โปรแกรม MATLAB® [Mathworks, 1991] ซึ่งเป็นซอฟต์แวร์ที่ใช้ง่าย มีประสิทธิภาพและเป็นที่นิยมแพร่หลายในวงวิศวกรรม ได้มีผู้พัฒนา GA Toolbox Chipperfield et al. [1994] ขึ้นมาเพื่อใช้กับโปรแกรม MATLAB® โดยผู้อ่านสามารถค้นหารายละเอียดเพิ่มเติมได้ที่

<http://www.shef.ac.uk/uni/projects/gaipp/gatbx.html>

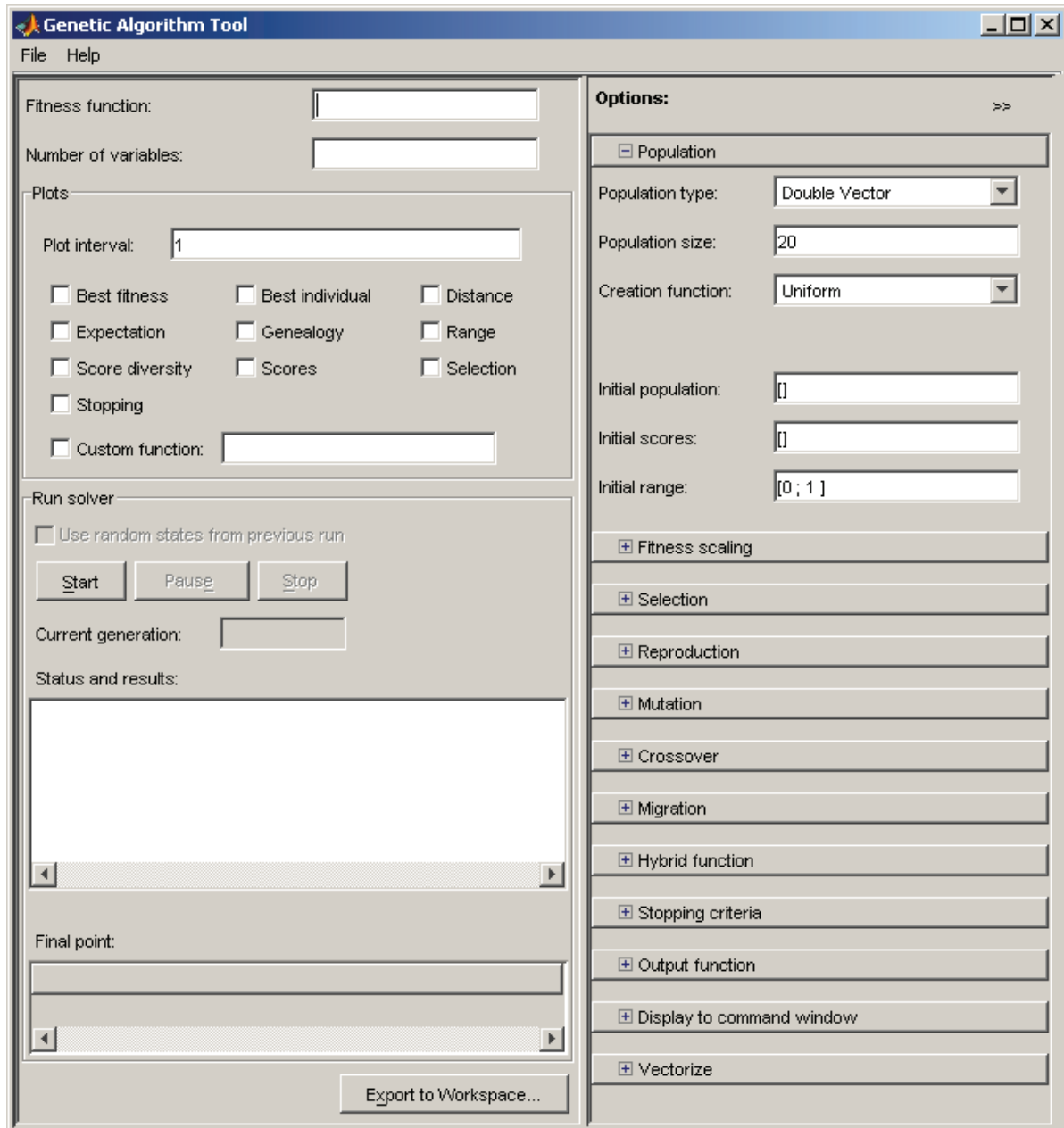
ส่วนในชุดซอฟต์แวร์ MATLAB® เวอร์ชัน 7.0 ขึ้นไปได้มีการพัฒนากล่องเครื่องมือ (Toolbox) ของ GA อยู่ในชื่อ *Genetic Algorithm and Direct Search* ซึ่งเป็นส่วนขยายมาจาก *Optimization Toolbox* ในหัวข้อต่อไปจะได้กล่าวถึงขั้นตอนและรายละเอียดการใช้งากล่องเครื่องมือของ GA สำหรับ MATLAB® รวมไปถึงตัวอย่างการใช้งานจริงสำหรับแสดงขั้นตอนการใช้งานแบบ GUI ของกล่องเครื่องมือดังกล่าว [MathWorks, 2004]

การใช้ Genetic Algorithm Tool

Genetic Algorithm Tool หรือ GAT เป็น GUI (Graphic User Interface) ที่ช่วยทำให้สามารถใช้ GA ได้โดยไม่ต้องใช้คำสั่งจากตัวพร้อมรับคำสั่ง (command line) ในการเปิดใช้งาน GAT ทำได้โดยการเรียกคำสั่งต่อไปนี้จากตัวพร้อมรับคำสั่งในหน้าต่าง MATLAB®

```
>> gatool
```

คำสั่งนี้จะทำการเปิดหน้าต่าง GAT ดังแสดงในรูปที่ 3.16 องค์ประกอบต่างๆ ของ GAT แสดงถึงพารามิเตอร์ของ GA ที่ได้นำเสนอมาในบทนี้ ซึ่งเป็นองค์ประกอบหลักๆ ของ GA ยกตัวอย่างเช่นช่อง Fitness function: ใช้สำหรับระบุฟังก์ชันค่าความเหมาะสมใน M ไฟล์ที่ต้องการหาค่าที่เหมาะสมที่สุด (ฟังก์ชันค่าความเหมาะสมใน GAT ดังกล่าวหมายถึงฟังก์ชันวัตถุประสงค์ในหัวข้อการประเมินค่าความเหมาะสม ตามที่ได้นำเสนอมาแล้วข้างต้นในบทนี้) ส่วนช่อง Number of variables: ใช้สำหรับระบุจำนวนตัวแปรในฟังก์ชันที่ต้องการหาค่าที่เหมาะสมที่สุด หรือส่วนต่างๆ ของ GUI ที่เป็นส่วนสำหรับปรับแต่งพารามิเตอร์ของ GA เช่นการทำครอสโอเวอร์และมิวเทชัน เป็นต้น ดังนั้นในการเริ่มใช้งาน GAT จึงจำเป็นจะต้องมีการระบุฟังก์ชันค่าความเหมาะสมและจำนวนตัวแปรในฟังก์ชันดังกล่าวก่อน ในที่นี้จะใช้ฟังก์ชัน peaks สำหรับสาธิตการทำงานของ GAT รูปที่ 3.17 แสดงลักษณะของจุดสูงสุดและต่ำสุดแบบวงแคบเฉพาะถิ่น (local) และแบบวงกว้าง (global) ของฟังก์ชัน peaks



รูปที่ 3.16: Genetic Algorithm Tool หรือ GAT

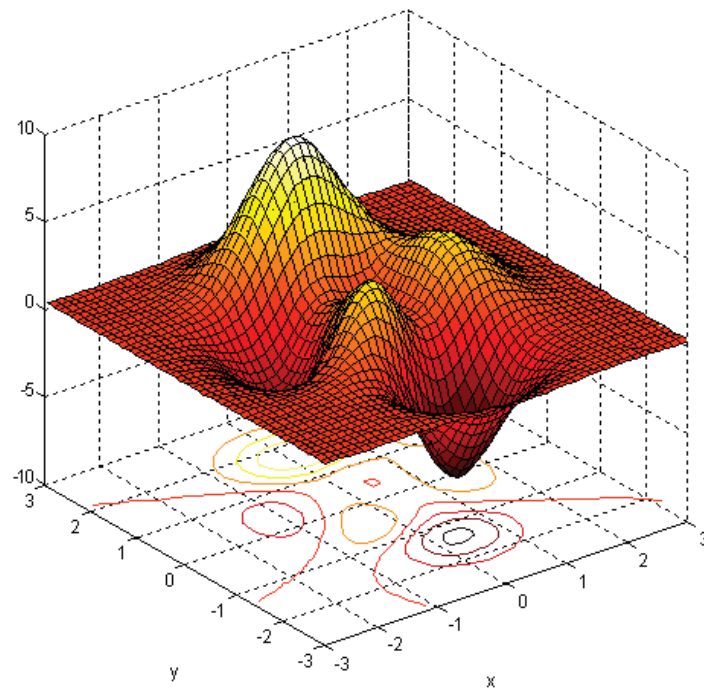
การเขียนฟังก์ชันค่าความเหมาะสมด้วยไฟล์ M

ฟังก์ชัน peaks เป็นฟังก์ชันสองตัวแปร สามารถเขียนได้เป็น

$$f(x, y) = 3(1 - x)^2 e^{-(x^2 + (y+1)^2)} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{-(x^2 + y^2)} - \frac{1}{3} e^{-((x+1)^2 + y^2)} \quad (3.9)$$

ดังนั้นไฟล์ M จะต้องรับตัวแปรเป็นเวกเตอร์แถว *pos* ที่มีขนาดเท่ากับ 2 สำหรับใช้แทนตัวแปร *x* และ *y* ในสมการข้างต้นและทำการส่งค่า *f(pos)* กลับ ไฟล์ M ของฟังก์ชันสามารถเขียนได้ดังนี้

```
function f=peaksfunction(pos)
f = 3*(1-pos(1)).^2.*exp(-(pos(1).^2) - (pos(2)+1).^2) ...
```



รูปที่ 3.17: ฟังก์ชัน peaks จาก MATLAB

```
- 10*(pos(1)/5 - pos(1).^3 - pos(2).^5).*exp(-pos(1).^2-pos(2).^2) ...
- 1/3*exp(-(pos(1)+1).^2 - pos(2).^2);
```

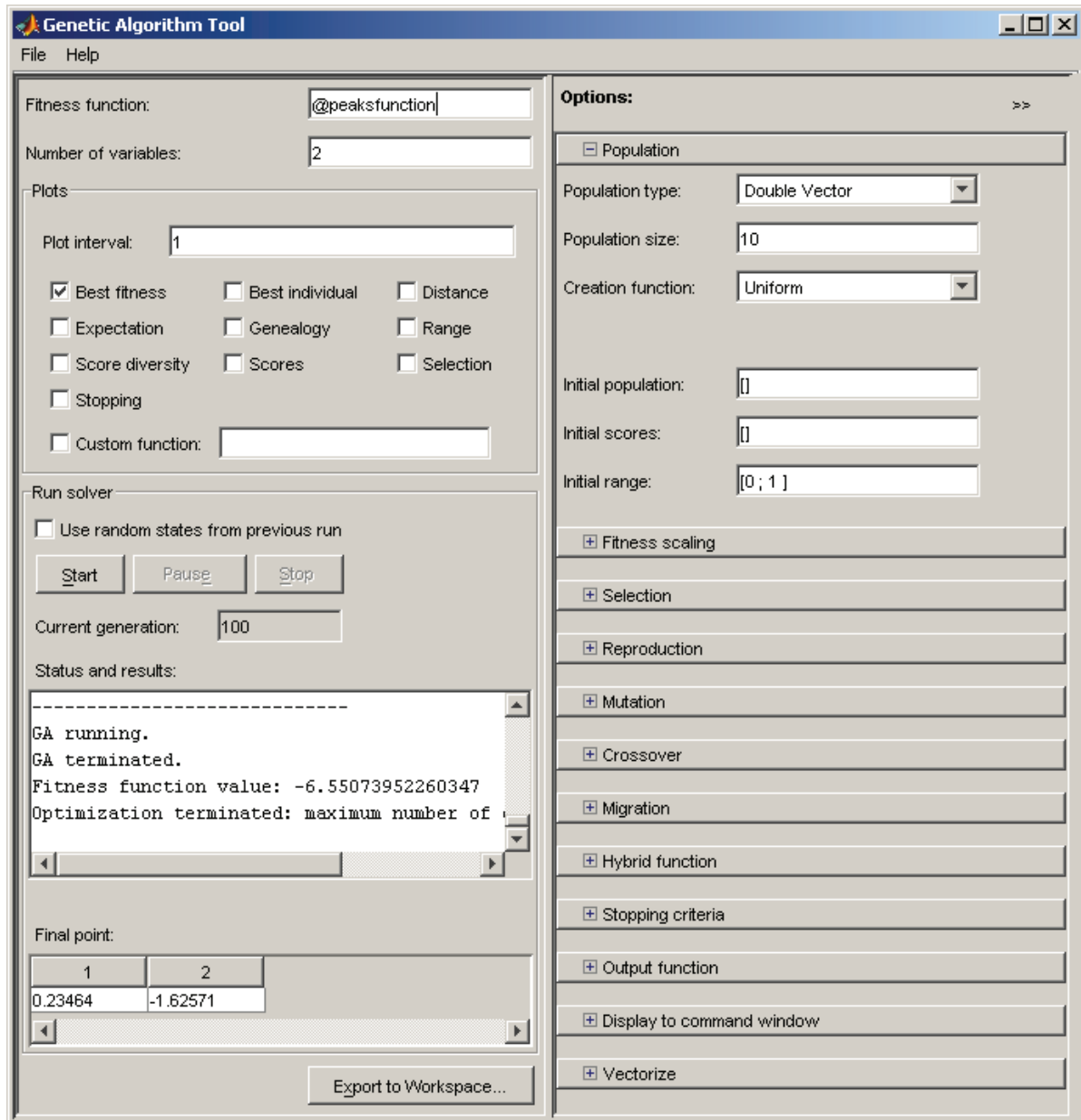
ในที่นี้ตัวแปร x และ y แทนด้วย $\text{pos}(1)$ และ $\text{pos}(2)$ ตามลำดับ เราจะทำการออกแบบ GA ในการค้นหาคำตอบในรูปของพิกัด (x, y) ที่ทำให้ฟังก์ชัน peak มีค่าสูงที่สุด ดังนั้นเราต้องทำการออกแบบ GA ที่มีจำนวนตัวแปรทั้งสิ้น 2 ตัวแปรคือ x และ y ใน GAT การหาค่าเหมาะที่สุดเป็นการหาค่าน้อยที่สุด (minimize) กล่าวคือ

$$\min_x f(x) \quad (3.10)$$

ถ้าเราต้องการหาค่าที่มากที่สุด เราสามารถทำได้โดยการหาค่าน้อยที่สุดของฟังก์ชัน $-f(x)$ นั่นเอง พารามิเตอร์อื่นๆ ที่กำหนดคือจำนวนรุ่นสูงสุดของ GA ที่จะทำการวนรอบคือ 100 รุ่น และจำนวนประชากร (หรือโครโมโซม) มีขนาดเท่ากับ 10

ผลการทดสอบ GAT

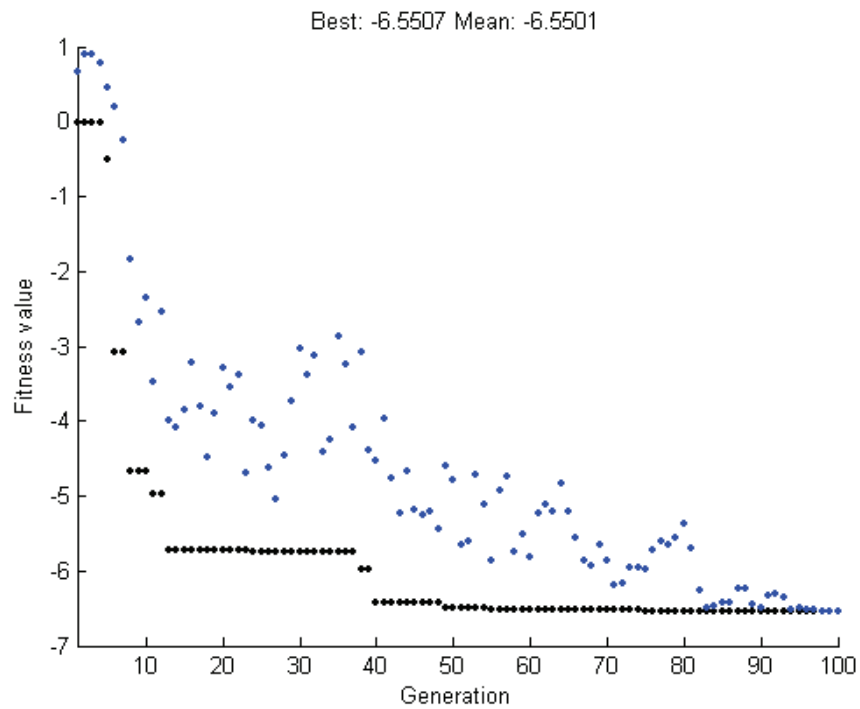
หลังจากทำการตั้งค่า GAT ตามฟังก์ชันที่ออกแบบไว้แล้ว ทำการกดปุ่ม **Start** เพื่อให้ GAT เริ่มทำงาน เราสามารถให้ GAT แสดงผลการทำงานในแต่ละรอบ (หรือทุกๆ จำนวนรอบที่ต้องการจากค่าใน `Plot interval`;) โดยสามารถเลือกรูปแบบการแสดงผลได้ เช่นค่าความเหมาะสมที่ดีที่สุดในแต่ละรอบหรือโครโมโซมที่ดีที่สุดในแต่ละรอบ เป็นต้น ผลการทำงานของ GAT สำหรับค้นหาค่าเหมาะที่สุดของฟังก์ชัน peaks แสดงในรูปที่ 3.18 จากผลที่ได้เราจะได้ว่าค่าต่ำสุดที่ GA หาได้คือ -6.55073952260347 โดยมีพิกัดคือ $(x, y) = (0.23464, -1.62571)$ กราฟค่าความเหมาะสมที่ดีที่สุดในแต่ละรอบแสดงในรูปที่ 3.19 สังเกตการลู่เข้าหาค่าตอบของค่าความเหมาะสมที่ดีที่สุดในแต่ละรอบ รูปที่ 3.20 แสดงกราฟช่วงระหว่างค่าความเหมาะสมที่ดีที่สุดและแย่ที่สุดในแต่ละรุ่น เส้นกราฟตรงกลางคือค่าเฉลี่ยของค่าความเหมาะสมของประชากรทั้งหมดในแต่ละรุ่น แนวโน้มช่วงของความเหมาะสมดังกล่าว



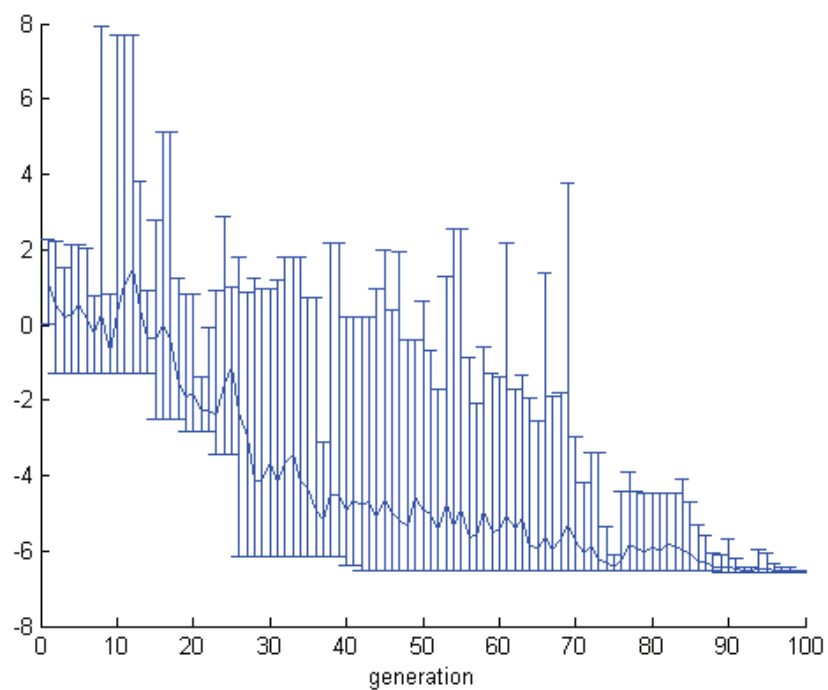
รูปที่ 3.18: ผลการทำงานของ GAT ในการค้นหาค่าน้อยที่สุดของฟังก์ชัน peaks

จะมีขนาดแคบลงในแต่ละรุ่นที่ผ่านไป ซึ่งแสดงให้เห็นถึงการลู่เข้าสู่คำตอบที่เป็นค่าที่น้อยที่สุด ส่วนรูปที่ 3.21 แสดงค่าเฉลี่ยระยะทางระหว่างโครโมโซมในแต่ละรอบ เป็นการแสดงให้เห็นว่าโครโมโซมใหม่ในแต่ละรุ่นมีการปรับปรุงสายพันธุ์ให้ดีขึ้น และการที่ระยะทางระหว่างแต่ละโครโมโซมลดลง แสดงว่าประชากรโครโมโซมทั้งหมดพยายามที่จะปรับตัวเข้าหาคำตอบด้วยกันทั้งหมด

ผลการค้นหาคำตอบที่ดีที่สุดแสดงให้เห็นถึงประสิทธิภาพของ GA โดยเฉพาะเมื่อพิจารณากระบวนการค้นหาที่ประกอบไปด้วยค่าความเหมาะสมแบบวงแคบเฉพาะถิ่นและแบบวงกว้าง ซึ่งโดยปกติแล้ว ปัญหาต่างๆ ไปจะมีลักษณะของพื้นผิวค่าความผิดพลาดหรือพื้นผิวสมรรถนะ (error surface หรือ performance surface) ที่ประกอบด้วยจุดวงแคบเฉพาะถิ่นและแบบวงกว้าง ลักษณะดังกล่าวมีผลต่อการค้นหาคำตอบ ที่ซึ่งระเบียบวิธีการหาค่าเหมาะสมที่สุดหลายๆ วิธีไม่สามารถที่จะเอาชนะปัญหาดังกล่าวได้ GA ได้รับการพิสูจน์และยอมรับถึงจุดเด่นใน

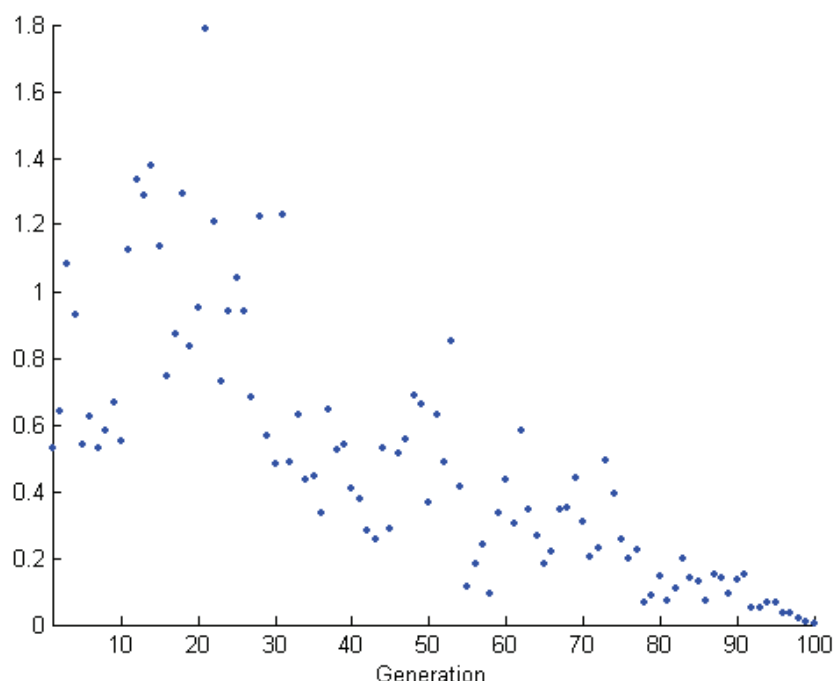


รูปที่ 3.19: ค่าความเหมาะสมที่ดีที่สุดในแต่ละรอบ



รูปที่ 3.20: ช่วงค่าที่ดีที่สุด ที่แย่ที่สุดและค่าเฉลี่ยของค่าความเหมาะสมในแต่ละรอบ

ข้อนี้ ทำให้เป็นวิธีที่น่าสนใจในการนำไปใช้งานได้หลากหลาย



รูปที่ 3.21: ค่าเฉลี่ยระยะทางระหว่างโครโมโซมในแต่ละรอบ

3.9 การประยุกต์ใช้งาน GA: การค้นหาสัมประสิทธิ์ของอนุกรมฟูรีเยร์

Genetic Algorithm หรือ GA เป็นขบวนการค้นหาคำตอบเชิงวิวัฒนาการวิธีหนึ่ง ที่มีประสิทธิภาพในการค้นหาคำตอบได้อย่างมีประสิทธิภาพ ด้วยโครงสร้างที่ไม่ซับซ้อน จึงทำให้ GA มีความเหมาะสมในการนำไปประยุกต์ใช้ในการค้นหาคำตอบสำหรับงานต่างๆ ได้เป็นอย่างดี หัวข้อนี้นำเสนอตัวอย่างการประยุกต์ใช้ GA มาช่วยในการคำนวณสัมประสิทธิ์ของอนุกรมฟูรีเยร์ [Srikaew, 2002] งานตัวอย่างนี้มีส่วนคล้ายคลึงกับงานของ Qi-Wen Yang [Yang et al., 2000] ที่ได้นำเอา GA มาประยุกต์ใช้ร่วมกับเครือข่ายประสาทเทียมในการค้นหาสัมประสิทธิ์ของอนุกรมฟูรีเยร์เช่นกัน โดยปกติแล้วการคำนวณหาสัมประสิทธิ์ดังกล่าว จะต้องเกี่ยวข้องกับการอินทิเกรตและใช้ขั้นตอนทางแคลคูลัสอื่นๆ มาช่วย ทำให้ในบางกรณีนั้นมีความยุ่งยากในการหาผลเฉลยในรูปแบบปิด การนำเอา GA มาช่วยในการคำนวณสัมประสิทธิ์ ทำให้ภาระในการคำนวณส่วนดังกล่าวลดลง เมื่อทำการออกแบบส่วนประกอบต่างๆ ของ GA รวมไปถึงการปรับค่าพารามิเตอร์ต่างๆ ให้เหมาะสมกับปัญหาแล้ว ระบบจะสามารถค้นหาคำตอบได้ในแบบอัตโนมัติ

3.9.1 อนุกรมฟูรีเยร์

อนุกรมฟูรีเยร์สามารถใช้แทนฟังก์ชันรายคาบ $f(x)$ ใดๆ ให้อยู่ในรูปของฟังก์ชันไซน์และโคไซน์ได้ กล่าวคือสำหรับฟังก์ชัน $f(x)$ ใดๆ ที่มีคาบ p เท่ากับ $2P$ จะได้ว่าอนุกรมฟูรีเยร์ของ $f(x)$ คือ [Kreyszig, 1999]

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left(a_n \cos \frac{n\pi}{P}x + b_n \sin \frac{n\pi}{P}x \right) \quad (3.11)$$

โดยที่สัมประสิทธิ์ฟูรีเยร์ของ $f(x)$ คือ

$$a_0 = \frac{1}{2P} \int_{-P}^P f(x) dx \quad (3.12)$$

$$\boxed{a_0 \quad a_1 \quad \dots \quad a_L \quad b_1 \quad b_2 \quad \dots \quad b_L}$$

รูปที่ 3.22: โครโมโซมสัมประสิทธิ์ของอนุกรมฟูรีเยร์

$$a_n = \frac{1}{P} \int_{-P}^P f(x) \cos \frac{n\pi x}{P} dx \quad (3.13)$$

$$b_n = \frac{1}{P} \int_{-P}^P f(x) \sin \frac{n\pi x}{P} dx \quad (3.14)$$

โดยที่ $n = 1, 2, \dots$ ดังนั้นส่วนที่ GA จะทำการค้นหาก็คือ $a_0, a_1, a_2, \dots, b_1, b_2, \dots$

3.9.2 GA กับอนุกรมฟูรีเยร์

การออกแบบให้ GA ทำการคำนวณหาสัมประสิทธิ์ของอนุกรมฟูรีเยร์มีส่วนที่ GA จะต้องสัมพันธ์กับอนุกรมฟูรีเยร์อยู่ 2 ส่วนคือ

- **โครโมโซม** - คือส่วนที่เป็นตัวแทนของคำตอบของระบบ ซึ่งคำตอบในที่นี้ก็คือสัมประสิทธิ์ของอนุกรมฟูรีเยร์ที่ต้องการให้ GA ค้นหา
- **ฟังก์ชันวัตถุประสงค์** - คือส่วนที่ใช้เชื่อมโยงระหว่างขบวนการค้นหาของ GA กับการคำนวณหาสัมประสิทธิ์การค้นหาของ GA จะต้องใช้ฟังก์ชันวัตถุประสงค์ในการตรวจสอบว่าโครโมโซมที่มีอยู่ในระบบนั้น มีความถูกต้องมากน้อยเพียงใด

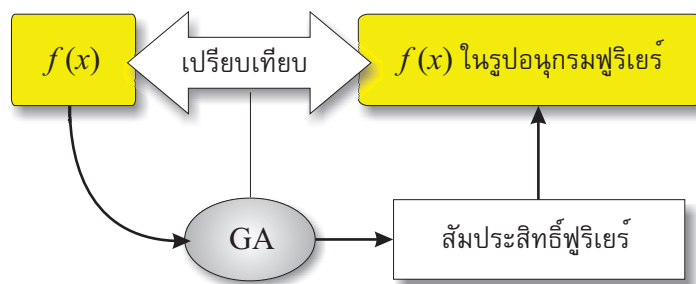
ดังนั้นเมื่อเริ่มต้นออกแบบใช้งาน GA ทั้งโครโมโซมและฟังก์ชันวัตถุประสงค์ถือว่าเป็นสององค์ประกอบสำคัญ ที่ต้องพิจารณาออกแบบก่อน องค์ประกอบทั้งสองเป็นส่วนที่ขึ้นอยู่กับชนิดของปัญหา ดังนั้นแต่ละระบบหรือแต่ละปัญหาจะมีรายละเอียดขององค์ประกอบดังกล่าวแตกต่างกันออกไป เนื่องจากว่าทั้งโครโมโซมและฟังก์ชันวัตถุประสงค์เป็นส่วนที่เชื่อมโยง GA เข้ากับระบบ การออกแบบเลือกรายละเอียดขององค์ประกอบทั้งสอง จึงมีผลโดยตรงต่อประสิทธิภาพของการค้นหาคำตอบ โดยมีรายละเอียดดังต่อไปนี้

3.9.3 โครโมโซม

โครโมโซมใน GA คือคำตอบที่ต้องการของระบบ ในที่นี้คำตอบของระบบที่ต้องการค้นหาคือสัมประสิทธิ์ของอนุกรมฟูรีเยร์ $a_0, a_1, a_2, \dots, b_1, b_2, \dots$ จะเห็นได้ว่าขนาดของโครโมโซมจะขึ้นอยู่กับจำนวนเทอมของสัมประสิทธิ์ที่ต้องการค้นหา ถ้ากำหนดให้จำนวนเทอมดังกล่าวคือ L ดังนั้นโครโมโซมที่ใช้จะมีขนาด $2L + 1$ (สัมประสิทธิ์ $a_0 - a_L$ และ $b_1 - b_L$) รูปที่ 3.22 แสดงสัมประสิทธิ์ของอนุกรมฟูรีเยร์ในรูปของโครโมโซมใน GA เราสามารถเขียนโครโมโซมในรูปของตัวแปรแบบสตริงได้ดังนี้

$$S = (a_0, a_1, \dots, a_L, b_1, b_2, \dots, b_L) \quad (3.15)$$

สำหรับการเข้ารหัสของโครโมโซมจะใช้วิธีเข้ารหัสแบบสายอักขระเลขฐานสอง (binary string) ส่วนจำนวนของโครโมโซมในแต่ละรุ่นนั้นจะกำหนดให้มีค่าเป็น N ตัว โดยจะทำการทดลองเปลี่ยนให้มีค่าต่างๆ เพื่อสังเกตถึงผลของขนาดของโครโมโซมต่อการค้นหาคำตอบของ GA



รูปที่ 3.23: ฟังก์ชันวัตถุประสงค์กับการเชื่อมโยงระบบเข้ากับ GA

3.9.4 ฟังก์ชันวัตถุประสงค์

ฟังก์ชันวัตถุประสงค์คือฟังก์ชันที่ใช้ในการประเมินค่าคำตอบที่ได้จากโครโมโซมในแต่ละรุ่น โดยเทียบกับวัตถุประสงค์หลักของระบบ ดังนั้นในกรณีของการค้นหาสัมประสิทธิ์ของอนุกรมฟูรีเยร์ จะต้องมีการประเมินว่าโครโมโซมหรือสัมประสิทธิ์ที่ได้จากการค้นหาด้วย GA นั้น ดีหรือไม่ดีอย่างไร การเชื่อมโยงระบบอนุกรมฟูรีเยร์เข้ากับ GA ด้วยฟังก์ชันวัตถุประสงค์แสดงในรูปที่ 3.23 วิธีหนึ่งที่ใช้ในการประเมินโครโมโซมคือการเปรียบเทียบค่าจริงของฟังก์ชัน $f(x)$ กับค่าประมาณของฟังก์ชัน $f(x)$ ที่สร้างจากสัมประสิทธิ์ของอนุกรมฟูรีเยร์ ซึ่งถ้าสัมประสิทธิ์ของอนุกรมฟูรีเยร์ที่ GA ทำการค้นหาได้มีความถูกต้อง ค่าของ $f(x)$ ที่สร้างจากสัมประสิทธิ์ชุดนั้นๆ ก็จะใกล้เคียงกับค่าจริงของ $f(x)$ กำหนดให้ค่าประมาณของฟังก์ชัน $f(x)$ ที่ได้จากอนุกรมฟูรีเยร์คือ

$$\hat{f}(x) = a_0 + \sum_{n=1}^L \left(a_n \cos \frac{n\pi}{P} x + b_n \sin \frac{n\pi}{P} x \right) \quad (3.16)$$

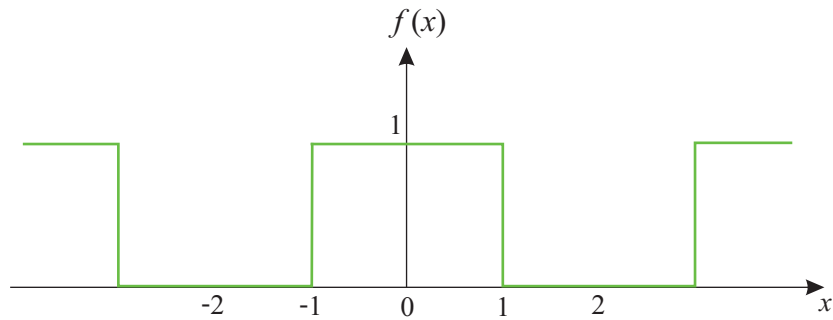
โดยที่ $a_0, a_1, a_2, \dots, b_1, b_2, \dots$ เป็นค่าสัมประสิทธิ์ที่ทำการถอดรหัสมาจากโครโมโซมชุดที่ดีที่สุดในแต่ละรุ่นของ GA การเปรียบเทียบระหว่าง $f(x)$ กับ $\hat{f}(x)$ จะให้ผลเป็นค่าการประเมินหรือค่าวัตถุประสงค์ (evaluation/objective value) ค่าดังกล่าวเป็นค่าที่ชี้ว่าสัมประสิทธิ์ที่ได้จากโครโมโซมสำหรับใช้ในการคำนวณ $\hat{f}(x)$ นั้นดีหรือไม่ดีอย่างไร วิธีที่นิยมใช้ทั่วไปในการเปรียบเทียบค่าประเมินดังกล่าว จะใช้การคำนวณหาความผิดพลาดระหว่าง $f(x)$ และ $\hat{f}(x)$ ในกรณีนี้จะใช้ค่าความผิดพลาดแบบผลรวมกำลังสอง e_f (sum-squared error) ภายในช่วงหนึ่งคาบของ $f(x)$ กล่าวคือ

$$e_f = \int_{-P}^P |f(x) - \hat{f}(x)|^2 dx \quad (3.17)$$

หรือ

$$e_f = \sum_{x=-P}^P |f(x) - \hat{f}(x)|^2 \quad (3.18)$$

ค่าความผิดพลาดในรูปของสมการที่ (3.18) นั้นเหมาะสำหรับใช้งานจริงในการคำนวณด้วยคอมพิวเตอร์ ค่าการชักตัวอย่างของตัวแปร x สามารถเลือกให้มีความละเอียดตามที่ต้องการ อย่างไรก็ตามความละเอียดดังกล่าวจะต้องเพียงพอ (นั่นคือความถี่ของการชักตัวอย่างจะต้องมากกว่าสองเท่าของความถี่ของฟังก์ชัน $f(x)$) และยังต้องเหมาะสมกับเวลาที่ใช้ในการคำนวณด้วยเช่นกัน (ความถี่ของการชักตัวอย่างที่สูงเกินไป ย่อมทำให้การคำนวณใช้เวลามากขึ้นด้วย) เมื่อใช้ค่าความผิดพลาดข้างต้นในการประเมินผลโครโมโซมจะได้ว่า โครโมโซมที่ดีกว่าจะให้ $\hat{f}(x)$ จากสัมประสิทธิ์ที่ให้ค่าความผิดพลาดที่น้อยกว่า ดังนั้น GA จะทำการค้นหาคำตอบในทิศทางที่ทำให้ค่าความผิดพลาดของระบบลดลง



รูปที่ 3.24: ฟังก์ชันตัวอย่าง $f(x)$ ในการคำนวณหาสัมประสิทธิ์ของอนุกรมฟูริเยร์

3.9.5 องค์ประกอบอื่นๆ ใน GA

นอกเหนือไปจากโครโมโซมและฟังก์ชันวัตถุประสงค์แล้ว องค์ประกอบอื่นๆ ในแต่ละขั้นตอนของ GA จะไม่ขึ้นอยู่กับชนิดของปัญหา ไม่ว่าจะเป็นฟังก์ชันกำหนดค่าความเหมาะสม การคัดเลือกสายพันธุ์ การทำครอสโอเวอร์ การทำมิวเทชัน และการแทนที่ รายละเอียดของขั้นตอนต่างๆ สามารถทำการทดลองออกแบบและปรับให้เหมาะสมกับระบบได้ ซอฟต์แวร์ที่ใช้ในการทดลองนี้คือ GA Toolbox สำหรับโปรแกรม MATLAB [Chipperfield et al., 1994] รายละเอียดของขั้นตอนต่างๆ จะได้กล่าวถึงในหัวข้อถัดไป

3.9.6 ฟังก์ชันตัวอย่างในการค้นหาสัมประสิทธิ์ของอนุกรมฟูริเยร์

ตัวอย่างฟังก์ชัน $f(x)$ ที่ใช้สำหรับคำนวณหาสัมประสิทธิ์ของอนุกรมฟูริเยร์แสดงในรูปที่ 3.24 ฟังก์ชัน $f(x)$ ที่ใช้ในที่นี้เป็นฟังก์ชันคู่ โดยกำหนดให้

$$f(x) = \begin{cases} 0, & -2 < x < -1 \\ 1, & -1 < x < 1 \\ 0, & 1 < x < 2 \end{cases} \quad (3.19)$$

โดยที่ $f(x)$ มีคาบเท่ากับ 4 หรือ $P = 2$ สำหรับ $\hat{f}(x)$ ในรูปของอนุกรมฟูริเยร์สามารถคำนวณได้ดังนี้ (ดูรูปที่ 3.25)

$$f(x) = \frac{1}{2} + \frac{2}{\pi} \left(\cos \frac{\pi}{2}x - \frac{1}{3} \cos \frac{3\pi}{2}x + \frac{1}{5} \cos \frac{5\pi}{2}x - \dots \right) \quad (3.20)$$

และสัมประสิทธิ์ของอนุกรมฟูริเยร์คือ

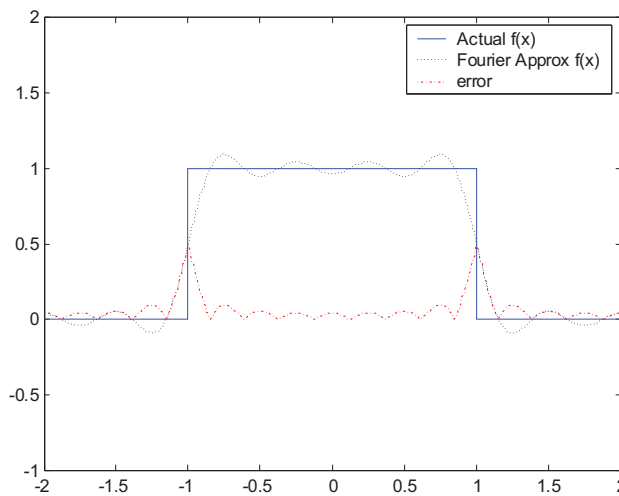
$$a_0 = \frac{1}{2} \quad (3.21)$$

$$a_n = \frac{2}{n\pi} \sin \frac{n\pi}{2} \quad (3.22)$$

$$b_n = 0 \quad (3.23)$$

ค่าเชิงตัวเลขของสัมประสิทธิ์ข้างต้นมีดังนี้

$a_0 = 0.500000$	$b_1 = 0.000000$
$a_1 = 0.636620$	$b_2 = 0.000000$
$a_2 = 0.000000$	$b_3 = 0.000000$
$a_3 = -0.212207$	$b_4 = 0.000000$
$a_4 = 0.000000$	$b_5 = 0.000000$
$a_5 = 0.127324$	$b_6 = 0.000000$
$a_6 = 0.000000$	$b_7 = 0.000000$
$a_7 = -0.090946$	

รูปที่ 3.25: ฟังก์ชันประมาณ $\hat{f}(x)$ จากอนุกรมฟูรีเยร์

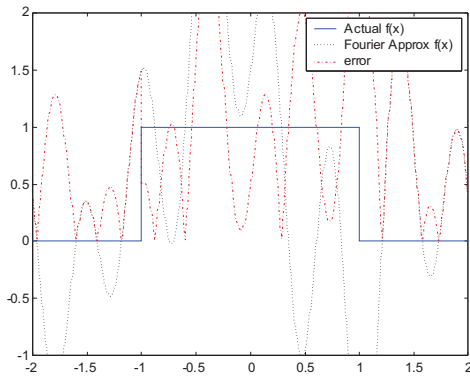
สังเกตว่าฟังก์ชัน $f(x)$ เป็นฟังก์ชันคู่ ทำให้เทอมของ b_n มีค่าเป็นศูนย์ ฟังก์ชัน $f(x)$ จึงเป็นตัวอย่างที่เหมาะสมกับการทดสอบว่า GA จะสามารถให้ผลของสัมประสิทธิ์ b_n ได้ถูกต้องหรือไม่ เพียงใดพิจารณาจำนวนเทอมของสัมประสิทธิ์ซึ่งเป็นตัวกำหนดขนาดของโครโมโซม จะสังเกตว่าเทอม a_n จะมีค่าเป็นศูนย์เมื่อ n เป็นเลขคู่ (ซึ่งทำให้เทอม $\sin n\pi/2 = 0$) ในกรณีตัวอย่างนี้จะใช้ $n = 7$ เพื่อให้จำนวนเทอมของสัมประสิทธิ์ a_n มีนัยสำคัญเพียงพอในสร้างฟังก์ชัน $\hat{f}(x)$ ให้ใกล้เคียงกับ $f(x)$ ดังนั้นแล้วขนาดของโครโมโซมในการทดลองนี้จะเท่ากับ 15 (นั่นคือ $a_0 - a_7$ และ $b_0 - b_7$)

3.9.7 ผลการค้นหาคำตอบ

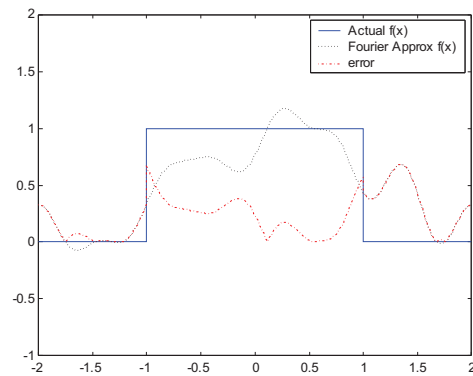
สำหรับฟังก์ชันวัตถุประสงค์ในการทดลองนี้ใช้การชักตัวอย่างของฟังก์ชัน $f(x)$ ในช่วงหนึ่งคาบเท่ากับ 800 จุด ค่าความผิดพลาดสูงสุด $error_{max}$ ระหว่าง $f(x)$ และ $\hat{f}(x)$ จากอนุกรมฟูรีเยร์มีค่าเท่ากับ 10.0887 รูปที่ 3.26 แสดงตัวอย่างของ $\hat{f}(x)$ ที่สร้างจากสัมประสิทธิ์ของโครโมโซมที่ดีที่สุดในแต่ละรุ่นในขณะที่ GA กำลังทำการค้นหาคำตอบ เนื่องจากขาบวนการของ GA เป็นขาบวนการที่มีการสุ่ม ในแต่ละรอบของการค้นหาคำตอบอาจจะได้ผลไม่เหมือนกัน ดังนั้นในตัวอย่างนี้จึงได้ทำการทดลองให้ GA ทำการค้นหาคำตอบเป็นจำนวนทั้งหมด 50 รอบแล้วพิจารณาคำตอบของระบบจากค่าเฉลี่ย องค์ประกอบต่างๆ ของ GA ที่ใช้ในการทดลองมีดังนี้

- การคัดเลือกสายพันธุ์แบบจัดอันดับ (ranking selection)
- การชักตัวอย่างแบบกระบวนการเฟ้นสุ่มครอบจักรวาล (stochastic universal sampling)
- การทำครอสโอเวอร์แบบจุดเดียวที่ระดับความน่าจะเป็นเท่ากับ 0.7
- การทำมิวเทชันที่ระดับความน่าจะเป็นเท่ากับ $\frac{0.7}{2L+1}$ (ขนาดของโครโมโซมเท่ากับ $2L + 1$)
- การแทนที่โครโมโซมที่มีค่าฟังก์ชันความเหมาะสมน้อยที่สุด (แทนที่เพียงรุ่นละหนึ่งโครโมโซม)

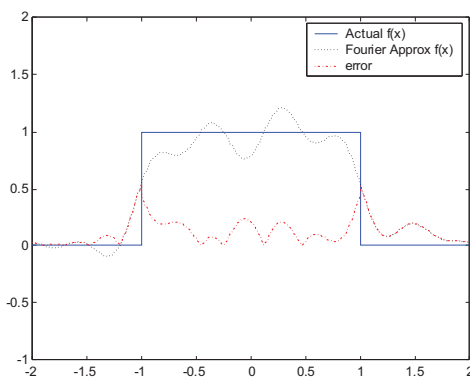
เมื่อค่าความผิดพลาดของโครโมโซมที่ดีที่สุดมีค่าน้อยกว่า $error_{max}$ ในแต่ละรอบแล้ว GA จะหยุดการค้นหาคำตอบ ทำการบันทึกคำตอบที่ได้ แล้วเริ่มทำการค้นหาคำตอบในรอบใหม่ จำนวนรุ่นเฉลี่ยในแต่ละรอบที่ GA ใช้ในการค้นหาคำตอบคือ 361 รุ่น ค่าเฉลี่ยของสัมประสิทธิ์อนุกรมฟูรีเยร์ที่ได้จาก GA มีดังนี้



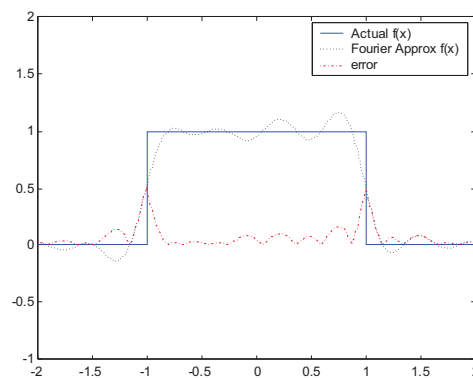
$\hat{f}(x)$ จากโครโมโซมเริ่มต้น (รุ่นที่ 0)



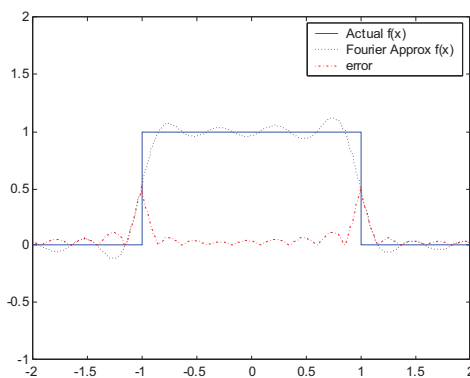
$\hat{f}(x)$ จากโครโมโซมรุ่นที่ 50



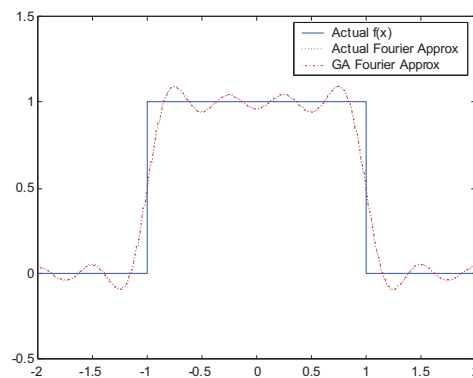
$\hat{f}(x)$ จากโครโมโซมรุ่นที่ 100



$\hat{f}(x)$ จากโครโมโซมรุ่นที่ 200



$\hat{f}(x)$ จากโครโมโซมรุ่นที่ 250



$\hat{f}(x)$ จากโครโมโซมรุ่นที่ 298 (รุ่นสุดท้าย)

รูปที่ 3.26: โครโมโซมที่ดีที่สุดของ GA (ในรูปฟีนไทป์) ในประชากรรุ่นต่างๆ

ตารางที่ 3.1: เปรียบเทียบค่าสัมประสิทธิ์จากการคำนวณและจากการค้นหาด้วย GA

	ค่าจริง	ค่าจาก GA	ค่าความผิดพลาด (%)
a_0	0.500000	0.501460	0.299
a_1	0.636620	0.638800	0.342
a_2	0.000000	-0.002660	0.418
a_3	-0.212207	-0.211640	0.009
a_4	0.000000	0.002981	0.468
a_5	0.127324	0.127720	0.006
a_6	0.000000	-0.003232	0.508
a_7	-0.090946	-0.090421	0.008
b_1	0.000000	-0.000312	0.001
b_2	0.000000	-0.000277	0.000
b_3	0.000000	0.000005	0.000
b_4	0.000000	0.000175	0.000
b_5	0.000000	-0.000009	0.000
b_6	0.000000	-0.000001	0.000
b_7	0.000000	-0.000837	0.131

$$a_0 = 0.501460$$

$$a_1 = 0.638800 \quad b_1 = -0.000312$$

$$a_2 = -0.002660 \quad b_2 = -0.000277$$

$$a_3 = -0.211640 \quad b_3 = 0.000005$$

$$a_4 = 0.002981 \quad b_4 = 0.000175$$

$$a_5 = 0.127720 \quad b_5 = -0.000009$$

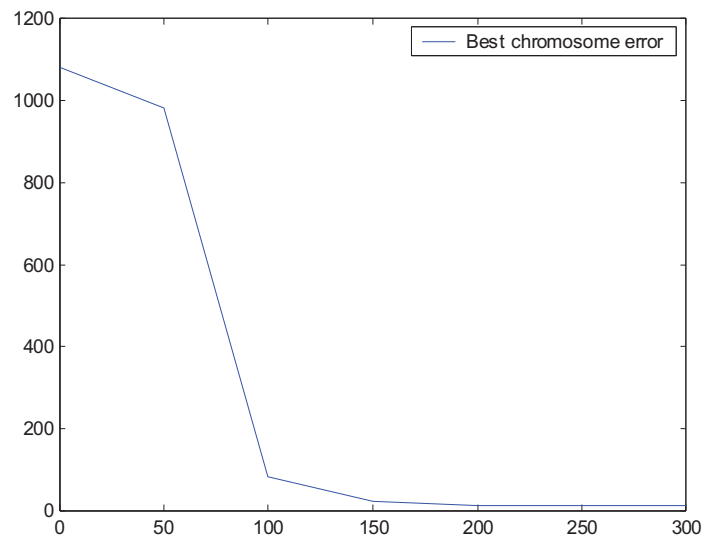
$$a_6 = -0.003232 \quad b_6 = -0.000001$$

$$a_7 = -0.090421 \quad b_7 = -0.000837$$

ตารางที่ 3.1 แสดงสัมประสิทธิ์ที่ได้จาก GA เมื่อเปรียบเทียบกับสัมประสิทธิ์ที่ได้จากการคำนวณ ค่าความผิดพลาดคิดเป็นเปอร์เซ็นต์เทียบกับค่า a_1 ซึ่งเป็นสัมประสิทธิ์ที่มีขนาดใหญ่ที่สุด จากตารางดังกล่าวจะเห็นได้ว่าค่าสัมประสิทธิ์ที่ได้จากการค้นหาด้วย GA มีค่าความผิดพลาดที่น้อยมาก การค้นหาสัมประสิทธิ์ของอนุกรมฟูรีเยร์ด้วย GA จึงถือว่ามีความถูกต้องแม่นยำสูง รูปที่ 3.27 แสดงตัวอย่างค่าความผิดพลาดกับการลู่เข้าของคำตอบที่ GA กำลังทำการค้นหา ซึ่งจะเห็นได้ว่าการลู่เข้าเป็นไปในทิศทางที่สม่ำเสมอจนกระทั่ง GA เจอคำตอบที่เหมาะสมในที่สุด

3.9.8 วิเคราะห์ผลการค้นหาคำตอบ

ผลการค้นหาสัมประสิทธิ์ฟูรีเยร์ที่ได้แสดงให้เห็นถึงรายละเอียดขั้นตอนการทำงาน และความสามารถในการค้นหาคำตอบของ GA อย่างมีประสิทธิภาพ อย่างไรก็ตาม GA มีองค์ประกอบภายในที่มีผลต่อประสิทธิภาพดังกล่าวด้วยเช่นกัน รายละเอียดขององค์ประกอบดังกล่าวมีเช่น ชนิดและขนาดของโครโมโซม กลวิธีในการคัดเลือกและแทนที่ประชากร พารามิเตอร์ของปฏิบัติการทางสายพันธุ์ เป็นต้นตัวอย่างการทดลองปรับเปลี่ยนจำนวนของโครโมโซมขนาดต่างๆ มีอยู่ในตารางที่ 3.2 ซึ่งแสดงผลการค้นหาคำตอบของ GA ที่ใช้จำนวนโครโมโซมต่างๆ กัน จากตารางจะเห็นได้ว่า ถ้าจำนวนโครโมโซมที่น้อยเกินไป จะทำให้ GA ต้องใช้เวลามากกว่าในการค้นหาคำตอบ (ใช้จำนวนรุ่นที่เจอคำตอบที่มากกว่า) ในขณะที่จำนวนโครโมโซมที่มากเกินไปนั้น ถึงแม้จะใช้จำนวนรุ่นในการค้นหาคำตอบ



รูปที่ 3.27: การลู่เข้าหาคำตอบจากค่าความผิดพลาดของโครโมโซมที่ดีที่สุดในแต่ละรุ่น

ตารางที่ 3.2: เปรียบเทียบผลที่ได้จากการใช้จำนวนโครโมโซมต่างๆ กัน

จำนวนโครโมโซมที่ใช้	20	40	50	60	100	150
จำนวนรุ่นที่เจอคำตอบ	839	458	361	335	245	223
ค่าความผิดพลาดเฉลี่ย	10.085	10.085	10.087	10.087	10.087	10.088

ที่น้อยกว่า แต่เวลาที่ใช้ในการคำนวณแต่ละรุ่นจะมากกว่า (เนื่องจากต้องทำการประเมินโครโมโซมในจำนวนที่มากขึ้น) ดังนั้นการกำหนดจำนวนโครโมโซม หรือแม้แต่พารามิเตอร์อื่นๆ ในแต่ละองค์ประกอบของ GA จะมีผลต่อการทำงานของระบบ การปรับพารามิเตอร์ต่างๆ เพื่อให้ GA มีประสิทธิภาพสูงสุดจึงมีความจำเป็นต่อระบบเสมอ GA ให้ผลการค้นหาคำตอบของระบบได้อย่างมีประสิทธิภาพ ทำให้ลดความยุ่งยากในขั้นตอนของการคำนวณหาผลเฉลยรูปแบบปิดของระบบ อย่างไรก็ตาม เนื่องมาจากขั้นตอนหลายๆ อย่างใน GA มีขั้นตอนการที่เป็นแบบสุ่ม ทำให้ในแต่ละครั้งของการค้นหาคำตอบ เวลาที่ใช้สำหรับปัญหาอย่างเดียวกันสามารถมีความแตกต่างกันได้ ดังนั้นถ้าพิจารณาเปรียบเทียบกับวิธีการหาคำตอบด้วยวิธีเชิงตัวเลข การค้นหาสัมประสิทธิ์ของอนุกรมฟูรีเยร์จะใช้เพียงการอินทิเกรตเชิงตัวเลขของฟังก์ชันในสมการที่ 3.13 3.14 และ 3.14 ซึ่งประกอบไปด้วยการอินทิเกรตของ $a_0 - a_L$ และ $b_1 - b_L$ ทั้งหมดเป็นจำนวน $2L + 1$ เทอม เนื่องจากการอินทิเกรตเชิงตัวเลขมีอัลกอริทึมที่มีความแน่นอนในเวลาที่ใช้และประสิทธิภาพของระบบ ดังนั้นจึงสามารถออกแบบระบบด้วยวิธีเชิงตัวเลข ให้มีผลการคำนวณที่เร็วและสามารถคาดคะเนผลการคำนวณได้แน่นอนกว่า ถึงแม้ว่าการใช้งานของระบบดังกล่าวจะมีความยุ่งยากกว่า GA ก็ตาม

3.9.9 สรุปการค้นหาสัมประสิทธิ์อนุกรมฟูรีเยร์ด้วย GA

การประยุกต์ใช้ GA ในการค้นหาสัมประสิทธิ์อนุกรมฟูรีเยร์มีความแม่นยำในเชิงตัวเลขค่อนข้างสูง ทำให้ช่วยลดภาระในการคำนวณหาผลเฉลยรูปแบบปิดของระบบ ซึ่งในที่นี้จะต้องใช้การอินทิเกรตเพื่อหาสัมประสิทธิ์ ในกรณี

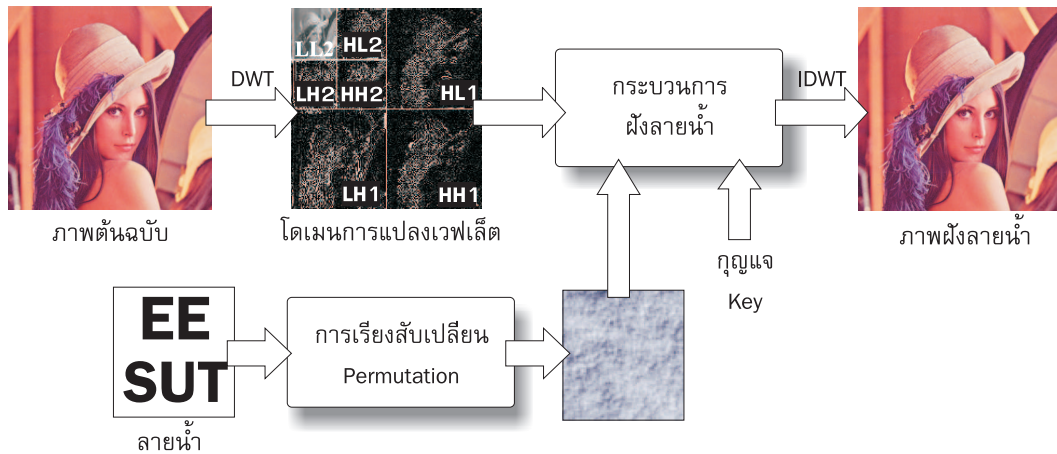
ที่ฟังก์ชัน $f(x)$ มีความซับซ้อน การคำนวณหาผลเฉลยรูปแบบปิดอาจจะมีความยุ่งยากมากยิ่งขึ้น และจากการออกแบบองค์ประกอบต่างๆ ของ GA จะเห็นได้ว่า การนำเอา GA มาค้นหาสัมประสิทธิ์ของอนุกรมฟูรีเยร์นั้นใช้เพียงข้อมูลชักตัวอย่างของฟังก์ชัน $f(x)$ เท่านั้น การนำเอา GA มาใช้จึงเป็นการเน้นในประสิทธิภาพการคำนวณพร้อมกับการลดภาระงานที่ยุ่งยากลงไป อย่างไรก็ตาม GA ที่ใช้ในการทดลองนี้ไม่ได้มีการปรับเปลี่ยนพารามิเตอร์ขององค์ประกอบภายในต่างๆ เช่น วิธีการคัดเลือกหรือแทนที่แบบต่างๆ วิธีโครอสโอเวอร์และมิวเทชันที่ระดับความน่าจะเป็นต่างๆ ฯลฯ ดังนั้นผลที่ได้จาก GA อาจจะไม่มีความแตกต่างไปบ้างสำหรับค่าพารามิเตอร์อื่นๆ โดยทั่วไปแล้ว ผลที่น่าสนใจสำหรับการค้นหาคำตอบด้วย GA คือ จำนวนรุ่นที่ใช้ในการหาคำตอบชุดหนึ่งๆ กล่าวคือถ้าออกแบบให้ GA ทำการค้นหาคำตอบได้ในจำนวนรุ่นที่น้อยกว่าและภายใต้ฟังก์ชันวัตถุประสงค์เดียวกัน นั้นหมายความว่าความน่าจะเป็นที่องค์ประกอบของ GA นั้นๆ มีความเหมาะสมกับระบบที่ใช้ ทำให้การค้นหาคำตอบเป็นไปได้เร็วขึ้น การประยุกต์ใช้ GA ในการค้นหาคำตอบจะมีประสิทธิภาพมากยิ่งขึ้นและนำไปสู่ระบบที่เป็นอัตโนมัติได้ การนำเอาวิธีการค้นหาคำตอบแบบปัญญาเชิงคำนวณมาประยุกต์ใช้ในงานทางวิศวกรรมอื่นๆ จึงเป็นหัวข้อที่น่าสนใจและเป็นที่น่าศึกษาค้นคว้าต่อไป

3.10 การประยุกต์ใช้งาน GA: การค้นหาพารามิเตอร์สำหรับกระบวนการทำลายน้ำภาพดิจิทัล

ในหัวข้อนี้จะได้นำเสนอตัวอย่างการประยุกต์ใช้ GA ในการค้นหาค่าสัมประสิทธิ์ของเวฟเลต (wavelet) ที่เหมาะที่สุดในการทำลายน้ำภาพดิจิทัล (digital image watermarking) เนื้อหาในหัวข้อนี้รวบรวมมาจากงานวิจัยหลายฉบับใน [Kumsawat, Attakitmongcol, and Srikaew, 2005a][Kumsawat, Attakitmongcol, Srikaew, and Sujitjorn, 2004a][Kumsawat, Attakitmongcol, and Srikaew, 2005b][Kumsawat, Attakitmongcol, Srikaew, and Sujitjorn, 2004b] งานดังกล่าวเป็นผลงานวิจัยของผู้เขียนและทีมคณาจารย์ในสาขาวิชาวิศวกรรมไฟฟ้าและสาขาวิชาวิศวกรรมโทรคมนาคม มหาวิทยาลัยเทคโนโลยีสุรนารี ภายใต้หน่วยวิจัยการประมวลผลภาพและสัญญาณดิจิทัล (Image & Signal Processing Research Unit) การทำลายน้ำในภาพดิจิทัลเป็นวิธีการป้องกันลิขสิทธิ์ของภาพดิจิทัลอย่างหนึ่ง การทำลายน้ำเป็นกระบวนการนำเอาข้อมูลที่บ่งบอกถึงความเป็นเจ้าของซ่อนไว้ในข้อมูลต้นฉบับ ซึ่งในที่นี้ก็คือภาพดิจิทัลนั่นเอง อย่างไรก็ตามวิธีการซ่อนข้อมูลดังกล่าวจะต้องเป็นไปตามวัตถุประสงค์ 2 ข้อของการทำลายน้ำดังต่อไปนี้

- **คุณภาพในการมองเห็น (perceptual quality)** การฝังลายน้ำลงในข้อมูลภาพ เป็นการเพิ่มข้อมูลให้กับภาพ ดังนั้นคุณภาพในการมองเห็นจะต้องถูกรักษาไว้ในระดับที่ต้องการ กล่าวคือลายน้ำที่ทำการฝังเข้าไปในภาพต้นฉบับจะต้องไม่กระทบต่อคุณภาพของภาพ (ในระดับหนึ่ง)
- **ความทนทานต่อการถูกโจมตี (robustness quality)** ลายน้ำที่ฝังไว้ในข้อมูลต้นฉบับจะต้องไม่สามารถถูกขจัดออกได้ ไม่ว่าจะโดยการต้องการขจัดออกโดยตรงโดยผู้ที่ไม่ใช่เจ้าของภาพต้นฉบับ หรือโดยการที่ภาพดิจิทัลถูกใช้งานแล้วถูกประมวลผล (โจมตี) ด้วยกระบวนการต่างๆ

ลายน้ำสำหรับภาพดิจิทัลนั้นสามารถทำได้ทั้งในโดเมนพื้นที่ (spatial domain) และโดเมนแปลง (transform domain) ซึ่งได้มีการศึกษาไว้มากมายที่ได้ยืนยันว่าการทำลายน้ำในโดเมนแปลงนั้น ให้ผลของลายน้ำที่มีความคงทนต่อสัญญาณรบกวน และต่อการประมวลผลสัญญาณภาพดิจิทัลพื้นฐานทั่วไปเช่นการย่อ/ขยายภาพ การจัดส่วนภาพ การปรับสภาพแสง ฯลฯ ได้ดีกว่าการทำลายน้ำในโดเมนพื้นที่ นอกจากนั้นแล้ว ลายน้ำควรจะมีความคงทนต่อการบีบอัดข้อมูลแบบต่างๆ ที่ซึ่งเป็นกระบวนการที่ถูกใช้งานกับภาพดิจิทัลมากที่สุดอย่างหนึ่งในปัจจุบัน การเพิ่มประสิทธิภาพในการทำลายน้ำดิจิทัลเป็นหัวข้อที่ยังคงได้รับความสนใจอยู่จนถึงปัจจุบัน กรรวิธีหนึ่งก็คือการนำเอาวิธีทางปัญญาเชิงคำนวณมาช่วย โดยมองกระบวนการทำลายน้ำภาพดิจิทัลเป็นปัญหาการหาค่าเหมาะที่สุด (optimization problem) ดังนั้นการนำเอาจินเนติกอัลกอริทึมหรือ GA มาประยุกต์ใช้งานจึงเป็นการช่วยในขั้นตอนการทำลายน้ำได้อย่างมีประสิทธิภาพ



รูปที่ 3.28: กระบวนการฝังลายน้ำในภาพดิจิทัล

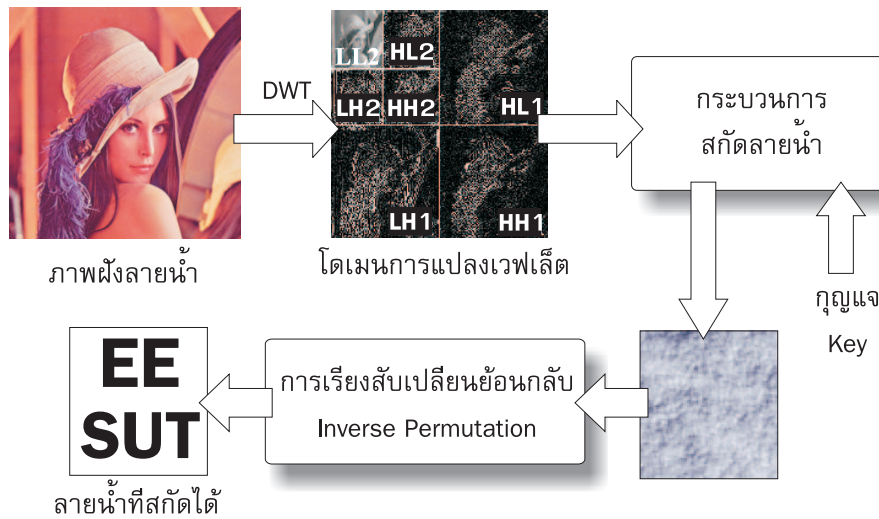
ในหัวข้อนี้ การแปลงเวฟเลตแบบดิสครีต (discrete wavelet transform หรือ DWT) ถูกใช้เป็นวิธีหลักในการทำลายน้ำในภาพดิจิทัล ผู้อ่านสามารถศึกษารายละเอียดของการทำลายน้ำด้วยวิธี DWT (และแบบมัลติเวฟเลต หรือ Discrete Multiwavelet Transform - DMT) และข้อมูลที่เกี่ยวข้องได้จากเอกสารอ้างอิงที่ได้กล่าวไว้ข้างต้น ในที่นี้ลายน้ำดิจิทัลจะถูกฝังในส่วนสัมประสิทธิ์ของเวฟเลต ที่ซึ่งมีขนาดมากกว่าค่าเริ่มเปลี่ยนที่ได้กำหนดไว้ก่อน GA ถูกนำมาใช้ในการค้นหาค่าพารามิเตอร์ของลายน้ำที่เหมาะสมที่สุด เพื่อให้ได้ลายน้ำที่มีประสิทธิภาพที่เหมาะสมที่สุดตามวัตถุประสงค์ที่ได้กำหนดไว้สำหรับการทำลายน้ำ โดยจะทำการค้นหาว่าสัมประสิทธิ์ส่วนใดของเวฟเลตที่ควรจะต้องถูกฝังลายน้ำ รายละเอียดต่างๆ ของกระบวนการฝังลายน้ำภาพดิจิทัล รวมทั้งหลักการแนวความคิดการนำเอา GA มาประยุกต์ใช้ในการค้นหาพารามิเตอร์ที่เหมาะสมที่สุด สำหรับการฝังลายน้ำมีดังต่อไปนี้

3.10.1 กระบวนการฝังลายน้ำภาพดิจิทัล

แผนผังกระบวนการทั่วไปในการฝังลายน้ำภาพดิจิทัลด้วยการแปลงเวฟเลตมีแสดงในรูปที่ 3.28 กระบวนการทั้งหมดเริ่มต้นที่การแปลงภาพต้นฉบับด้วยเวฟเลต (DWT หรือ DMT) การแปลงดังกล่าวจะมีความแตกต่างกันตามชนิดของเวฟเลตแม่ที่ใช้ รวมไปถึงระดับการแปลงซึ่งสามารถมีได้หลายแบบน้อย (ในรูปแสดงการแปลง 2 ระดับ ระดับที่ 1 มีแบนด์ย่อยคือ HL1 LH1 และ HH1 ระดับที่ 2 มีแบนด์ย่อยคือ LL2 HL2 LH2 และ HH2) ลายน้ำจะถูกรวมเข้ากับภาพที่ผ่านการแปลงนี้โดยการบวกเข้ากับสัมประสิทธิ์การแปลงของเวฟเลต เพื่อเป็นการเพิ่มประสิทธิภาพในการฝังลายน้ำ ก่อนการบวกลายน้ำเข้ากับสัมประสิทธิ์ดังกล่าวสามารถมีขั้นตอนการประมวลผลเพิ่มเติมได้ เช่นการเรียงสับเปลี่ยน (permutation) และสัญญาณรบกวนรูปแบบสุ่มเทียม (pseudo-random noise) ฯลฯ เข้าไปในสัญญาณลายน้ำพร้อมกับกุญแจ (key) เพื่อป้องกันการขจัดลายน้ำ (ผู้ที่มีกุญแจถึงจะขจัดลายน้ำออกได้) เป็นต้น สังเกตว่าในกรณีนี้สัญญาณลายน้ำมีลักษณะเป็นสัญญาณ 2 มิติ (นั่นคือเป็นรูปภาพดิจิทัล) ถ้ากำหนดให้ $I(u, v)$ เป็นสัมประสิทธิ์ DWT ที่ได้จากการแปลง DWT ของภาพต้นฉบับ $W(u, v)$ เป็นสัญญาณลายน้ำและ α คือตัวประกอบอัตราการขยายหรือความแรงของลายน้ำ จะได้ว่าสัมประสิทธิ์ DWT ของภาพที่ถูกฝังลายน้ำ $I_W(u, v)$ คือ

$$I_W(u, v) = \begin{cases} I(u, v) + \alpha W(u, v) & \text{โดยที่ } u, v \in HL1, HL1, HL2, LH2 \\ I(u, v) & \text{โดยที่ } u, v \in LL2, HH1, HH2 \end{cases}$$

สังเกตว่ามีการฝังลายน้ำในบางระดับของสัมประสิทธิ์ DWT เท่านั้น ซึ่งได้แก่ HL1 LH1 HL2 และ LH2 เมื่อได้สัมประสิทธิ์ที่ผ่านการฝังลายน้ำแล้ว เราสามารถแปลงสัมประสิทธิ์ DWT กลับ (inverse DWT หรือ IDWT) เพื่อได้ภาพที่มีการฝังลายน้ำในโดเมนเชิงพื้นที่หรือภาพปกติที่เรามองเห็นทั่วทั้งไปนั่นเอง



รูปที่ 3.29: กระบวนการสกัดลายน้ำจากภาพดิจิทัล

3.10.2 กระบวนการสกัดลายน้ำออกจากภาพดิจิทัล

กระบวนการสกัดลายน้ำออกจากภาพดิจิทัลเป็นกระบวนการย้อนกลับของกระบวนการฝังลายน้ำ รูปที่ 3.29 แสดงกระบวนการสกัดลายน้ำดังกล่าว ขั้นตอนแรกจะเป็นการคำนวณหาสัมประสิทธิ์ของเวฟเลต DWT จากภาพที่มีการฝังลายน้ำ สัญญาณลายน้ำที่ผ่านการเรียงสับเปลี่ยนในขั้นตอนการฝังลายน้ำจะถูกสกัดออกจากสัมประสิทธิ์ DWT นี้ โดยการวิเคราะห์สัมประสิทธิ์และสหสัมพันธ์ (correlation) ของอนุกรมแบบสุ่มเทียมที่ใช้ในขั้นตอนการสร้างลายน้ำ หลังจากนั้นจะทำการเรียงสับเปลี่ยนย้อนกลับสัญญาณลายน้ำเพื่อสกัดลายน้ำออกจากสัมประสิทธิ์ ในกรณีนี้สัญญาณลายน้ำเป็นสัญญาณภาพ (ตัวอย่างเช่นเป็นตัวอักษร EE SUT) เราสามารถประเมินความคล้ายของลายน้ำที่สกัดได้กับลายน้ำต้นฉบับด้วยค่าบรรทัดฐานของสหสัมพันธ์ (normalized correlation หรือ NC) ตามความสัมพันธ์ต่อไปนี้

$$NC = \frac{\sum_i \sum_j W(i,j) \hat{W}(i,j)}{\sum_i \sum_j W(i,j)^2}$$

โดยที่ $W(i,j)$ และ $\hat{W}(i,j)$ คือสัญญาณลายน้ำต้นฉบับและสัญญาณลายน้ำที่สกัดได้ตามลำดับ สังเกตว่าในกระบวนการสกัดลายน้ำนั้น เราไม่จำเป็นต้องใช้รูปภาพต้นฉบับแต่อย่างใด

3.10.3 การเพิ่มประสิทธิภาพการฝังลายน้ำด้วยจินเนติกอัลกอริทึม

ปกติแล้วกระบวนการฝังลายน้ำมีเป้าหมายอยู่ 3 อย่างที่มักจะขัดแย้งกันอยู่เสมอ เป้าหมายดังกล่าวก็คือความเนียน (imperceptibility) ความคงทน (robustness) และความจุข้อมูล (data capacity) ความเนียนคือการที่สัญญาณลายน้ำไม่สามารถมองเห็นด้วยตาเปล่าจากภาพที่ผ่านการฝังลายน้ำ กล่าวคือสัญญาณลายน้ำไม่ควรจะปรากฏในภาพที่ผ่านการฝังลายน้ำนั้นๆ ความคงทนคือการที่สัญญาณลายน้ำสามารถทนต่อการประมวลผลแบบต่างๆ ต่อภาพที่มีการฝังลายน้ำนั้นๆ ความจุข้อมูลเป็นผลมาจากปริมาณบิตของลายน้ำที่ต้องฝังลงในภาพต้นฉบับ จำนวนบิตที่มากแสดงถึงปริมาณข้อมูลเนื้อหาของสัญญาณลายน้ำที่มากขึ้น (เช่นภาพลายน้ำชนิดขาวดำใช้ข้อมูล 1 บิตต่อจุดภาพ ในขณะที่ภาพลายน้ำชนิดระดับเทาใช้ข้อมูล 8 บิตต่อจุดภาพ เป็นต้น) แต่ก็ทำให้ข้อมูลภาพต้นฉบับสูญเสียไปมากขึ้นด้วย ปริมาณหรือความแรงของสัญญาณลายน้ำที่มากทำให้การสกัดลายน้ำเป็นไปอย่างสะดวก แต่คุณภาพของภาพต้นฉบับจะแย่ลงเนื่องจากปริมาณของลายน้ำจะไปลดทอนสัมประสิทธิ์ DWT ของภาพต้นฉบับ ดังนั้นเพื่อเป็นการลดความขัดแย้งดังกล่าวให้มากที่สุด เราสามารถใช้ GA ในการค้นหาพารามิเตอร์

การฝังลายน้ำที่มีค่าเหมาะที่สุดได้ GA จะถูกประยุกต์ใช้ทั้งในขั้นตอนการฝังและการสกัดลายน้ำ โดยพารามิเตอร์ที่จะถูกค้นหาด้วย GA คือค่าความแรงของสัญญาณลายน้ำ α นั้นเอง ฟังก์ชันวัตถุประสงค์ของ GA จะทำการประเมินค่าจากปริมาณวัดของความคงทนและความเนียนของภาพที่ฝังลายน้ำ รายละเอียดตัวอย่างการใช้ GA ในการค้นหาสัมประสิทธิ์ α ที่เหมาะที่สุดในการฝังลายน้ำมีต่อไปนี้

- โครโมโซมของ GA ใช้แทนพารามิเตอร์ α
- จำนวนโครโมโซมเท่ากับ 20
- การเข้ารหัสเป็นแบบสายอักขระเลขฐานสองขนาด 32 บิต
- ฟังก์ชันวัตถุประสงค์มีตัวชี้วัดประสิทธิภาพของการฝังและสกัดลายน้ำจากปริมาณต่อไปนี้
 - ตัวชี้วัดคุณภาพสากล (Universal Quality Index หรือ UQI) - เป็นตัวชี้วัดคุณภาพของภาพที่ผ่านการฝังลายน้ำ ซึ่งได้แก่ความเนียน (imperceptibility)
 - ค่าบรรทัดฐานของสหสัมพันธ์ (Normalized Correlation หรือ NC) - เป็นปริมาณชี้วัดคุณภาพของสัญญาณลายน้ำที่สกัดได้ ซึ่งได้แก่ความคงทน (robustness)
- ฟังก์ชันวัตถุประสงค์เป็นไปตามความสัมพันธ์ต่อไปนี้

$$W = \delta_{UQI} \times UQI + \delta_{NC} \times NC$$

โดยที่ δ_{UQI} และ δ_{NC} เป็นตัวประกอบถ่วงน้ำหนักของตัวชี้วัด UQI และ NC ตามลำดับ ตัวประกอบถ่วงน้ำหนักนี้เป็นส่วนที่ระบุความสำคัญของตัวชี้วัด ถ้าตัวชี้วัดทั้งสองมีนัยสำคัญเท่ากัน ค่าของตัวประกอบถ่วงน้ำหนักทั้งสองจะมีเท่ากับ 0.5 ซึ่งต้องเป็นไปตามความสัมพันธ์ $\delta_{UQI} + \delta_{NC} = 1.0$

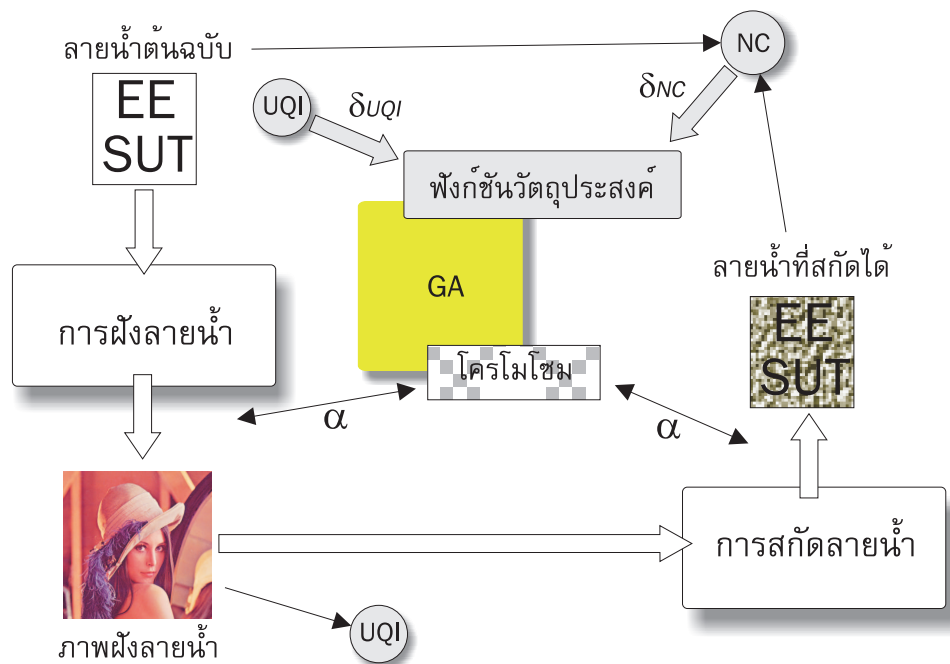
- พารามิเตอร์อื่นๆ ของ GA ใช้ตามค่าปกติได้แก่การคัดเลือกแบบจัดลำดับ การทำครอสโอเวอร์และมิวเทชันด้วยความน่าจะเป็นเท่ากับ 0.7 และ 0.05 ตามลำดับ และในแต่ละรุ่นประชากร โครโมโซมจะถูกแทนที่แบบบางส่วนเท่านั้น

รูปที่ 3.30 แสดงแผนผังการค้นหาพารามิเตอร์การฝังลายน้ำด้วย GA จะเห็นได้ว่า GA ถูกใช้งานทั้งในขั้นตอนการฝังลายน้ำและการสกัดลายน้ำ โดยทำการเลือกค่าพารามิเตอร์ α ที่เหมาะที่สุดให้กับระบบ

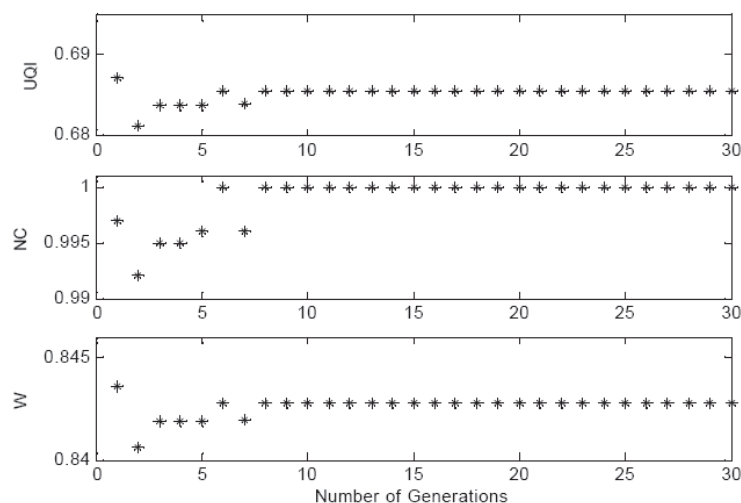
3.10.4 ผลการค้นหาพารามิเตอร์การฝังลายน้ำด้วย GA

รูปภาพขนาด 512×512 จุดภาพถูกใช้ในการทดสอบการฝังและสกัดลายน้ำ พร้อมกับภาพสัญญาณลายน้ำแบบไบนารีเป็นรูปสัญลักษณ์ตัวอักษร EE SUT ขนาด 32×32 จุดภาพ การใช้สัญญาณลายน้ำดังกล่าวสามารถมองเห็นได้ด้วยตาเปล่า ซึ่งในงานวิจัยที่ใช้สัญญาณรูปแบบอื่นที่ไม่จำเป็นจะต้องมองเห็นแล้วเข้าใจความหมาย เช่น สัญญาณสุ่ม เป็นต้น โดยปกติแล้วจุดประสงค์ของการฝังลายน้ำในที่นี้ก็คือสามารถตรวจจับว่า ‘มี’ หรือ ‘ไม่มี’ สัญญาณลายน้ำอยู่ภายในภาพเท่านั้น

รูปที่ 3.31 แสดงการลู่เข้าของ GA ณ ประชากรรุ่นที่ 30 จากการทดสอบกับภาพ Lena ผลของพารามิเตอร์ที่ GA ค้นหาได้สำหรับภาพทดสอบต่างๆ แสดงในตารางที่ 3.3 ตัวอย่างการสกัดสัญญาณลายน้ำมีแสดงในรูปที่ 3.32 ซึ่งจะเห็นว่าถึงแม้ว่าภาพฝังลายน้ำจะถูกบีบอัดแบบ JPEG สัญญาณลายน้ำที่สกัดได้ยังคงมองเห็นข้อมูลของต้นฉบับอยู่ ผลลัพธ์ที่ได้แสดงถึงการนำเอา GA มาประยุกต์ใช้กับงานทางด้านการฝังลายน้ำในภาพดิจิทัล ถ้าผู้อ่านสนใจผลการทดสอบระบบในเรื่องความเนียนของภาพฝังลายน้ำและความคงทนของลายน้ำ สามารถศึกษาเพิ่มเติมได้ในงานต้นฉบับตามเอกสารอ้างอิงข้างต้น



รูปที่ 3.30: การค้นหาพารามิเตอร์การผั่งลายหน้าด้วย GA

รูปที่ 3.31: การลู่เข้าของ UQI , NC และ W ในขั้นตอนการค้นหาด้วย GA (จาก [Kumsawat, Attakitmongcol, and Srikaew, 2005b])

3.11 สรุป

ข้อดีของ GA

เมื่อพิจารณาถึงความสามารถและโครงสร้างของ GA แล้ว สามารถสรุปข้อดีต่างๆ ของ GA ได้ดังนี้

- มีโครงสร้างที่เหมาะสมสำหรับการประมวลผลแบบขนาน โดยโครงสร้างของ GA แล้วสามารถถูกออกแบบให้ทำงานในลักษณะของการประมวลผลแบบขนานได้จึงสามารถทำให้ความเร็วในการคำนวณเพิ่มขึ้น GA สามารถแบ่งการคำนวณเป็นหน่วยย่อย โดยที่แต่ละหน่วยเป็น GA ที่ทำงานได้สมบูรณ์ในตัวเองและแยกกัน

ตารางที่ 3.3: พารามิเตอร์ α จากการค้นหาของ GA พร้อมทั้งค่า PSNR UQI และ NC จากภาพฝัງลายน้ำ (จาก [Kumsawat, Attakitmongcol, and Srikaew, 2005b])

ภาพ	α	PSNR (dB)	UQI	NC
Lena	0.6607	34.55	0.6885	1.0000
Barbara	0.6289	34.66	0.8279	0.8121
Baboon	0.6392	33.94	0.9408	0.5942
Gold Hill	0.6078	34.56	0.8590	0.9292
Boat	0.6115	35.22	0.8463	0.9292
Pepper	0.5997	34.37	0.7823	0.9290



(ก)



(ข)



(ค)

รูปที่ 3.32: (ก) ภาพฝัງลายน้ำ (ข) ภาพลายน้ำต้นฉบับ (ค) ภาพลายน้ำที่สกัดได้หลังจากภาพฝัງลายน้ำถูกบีบอัดแบบ JPEG ด้วยตัวประกอบคุณภาพภาพ 30% (จาก [Kumsawat, Attakitmongcol, and Srikaew, 2005b])

ทำการค้นหาคำตอบของระบบพร้อมๆ กันได้ นอกไปจากนี้แล้ว GA ยังสามารถแบ่งตัวเองเป็นหน่วยย่อยที่แต่ละหน่วยแยกทำหน้าที่ตามขั้นตอนของ GA ต่างๆ ได้ เช่น แยกเป็นหน่วยคำนวณค่าความเหมาะสม หน่วยทำการคัดเลือก หน่วยทำปฏิบัติการทางสายพันธุ์ เป็นต้น

- มีเสถียรภาพและความน่าเชื่อถือ เป็นที่ยืนยันแล้วว่า มีเทคนิคหลายๆ อย่างที่สามารถใช้ทำให้ GA มีการลู่เข้าหาคำตอบที่ดีที่สุดได้ถึงแม้ว่าสถานะแวดล้อมของระบบจะมีการเปลี่ยนแปลงไป
- สามารถหาคำตอบที่เหมาะสมที่สุดแบบวงกว้าง (global optimum) ปัญหาหลายๆ อย่างจะมีคำตอบอยู่หลายชุด ซึ่งเป็นคำตอบที่เหมาะสมที่สุดแบบวงแคบเฉพาะถิ่น (local optimum) GA ได้รับการพิสูจน์ให้เห็นว่าสามารถเอาชนะปัญหาของระบบดังกล่าวได้อย่างมีประสิทธิภาพ
- มีความยืดหยุ่นในการออกแบบชนิดของโครโมโซม ยีนของโครโมโซมสามารถถูกออกแบบให้เป็นตัวเลขชนิดใดๆ ก็ได้ตามความเหมาะสมกับระบบไม่ว่าจะเป็นแบบฐานสองหรือเลขจำนวนจริง จึงทำให้ GA สามารถใช้งานกับระบบต่างๆ ได้หลากหลาย
- เหมาะสำหรับระบบที่มีเงื่อนไขข้อจำกัดต่างๆ ในหลายๆ ระบบจะมีการระบุเงื่อนไขของพารามิเตอร์ต่างๆ เพื่อจำกัดให้ระบบมีเสถียรภาพและเป็นไปตามเงื่อนไขทางกายภาพ เงื่อนไขดังกล่าวถือเป็นการจำกัดพื้นที่

สำหรับการค้นหาคำตอบ GA มีโครงสร้างที่สามารถออกแบบให้มีการจำกัดขอบเขตของโครโมโซมได้อย่างสะดวก จึงเหมาะกับการแก้ไขปัญหาที่ต้องมีเงื่อนไขแบบต่างๆ ได้เป็นอย่างดี

- เหมาะสำหรับระบบที่มีฟังก์ชันวัตถุประสงค์แบบพหุคูณ GA สามารถใช้กับระบบที่มีฟังก์ชันวัตถุประสงค์มากกว่าหนึ่ง ยกตัวอย่างเช่น การออกแบบตัวกรองชนิด IIR ที่ต้องมีการกำหนดฟังก์ชันวัตถุประสงค์สำหรับทั้งค่าความผิดพลาดของขนาด (magnitude error) และค่าความผิดพลาดของเวลาประวิง (delay error) ลักษณะดังกล่าวทำให้ GA มีความเหมาะสมในการแก้ปัญหาดังกล่าวซึ่งส่วนใหญ่จะมีเป้าหมายในการแก้ปัญหามากกว่าหนึ่งอย่าง
- ใช้แก้ปัญหาของระบบได้โดยไม่ต้องรู้หรือคำนวณหาผลเฉลยรูปแบบปิด (closed form solution) ของระบบ โดยปกติแล้วการคำนวณหาผลเฉลยดังกล่าวจะมีความยุ่งยากและต้องใช้เวลามาก การนำเอา GA มาประยุกต์ใช้งานจึงช่วยลดความยุ่งยากในส่วนนี้ได้เป็นอย่างดี

ข้อจำกัดของ GA

GA มีข้อจำกัดในการใช้งานจริงเหมือนกัน ปัญหาหลักๆ ของ GA คือการปรับพารามิเตอร์ของ GA ที่ขึ้นอยู่กับปัญหาที่จะนำเอา GA ไปแก้ไขหาคำตอบ ผู้ใช้จะต้องมีความเข้าใจในปัญหา ซึ่งปัญหาดังกล่าวมีรายละเอียดความหลากหลายมากมาย แล้วเลือกปรับ GA ให้เข้ากับสภาวะแวดล้อมของปัญหาให้ได้ที่สุด อย่างไรก็ตาม GA ยังมีข้อจำกัดที่ผู้ใช้จะต้องเข้าใจดังตัวอย่างต่อไปนี้

- **การลู่เข้าก่อนวัย (premature convergence)** ปัญหาของการลู่เข้าก่อนวัยสามารถเกิดขึ้นจากการขาดความหลากหลาย (diversity) ของประชากรในระบบ เมื่อโครโมโซมส่วนใหญ่ในกลุ่มประชากรมีคุณลักษณะที่คล้ายๆ กัน กลไกการคัดเลือกสายพันธุ์จะไม่สามารถเลือกโครโมโซมที่เด่นได้ ส่งผลให้อัตราการเพิ่มขึ้นของค่าความเหมาะสมที่ลดลง การลู่เข้าก่อนวัยสามารถถูกตรวจจับได้จากการเปรียบเทียบค่าความเหมาะสมเฉลี่ยของประชากรกับค่าความเหมาะสมที่ดีที่สุด ถ้าค่าทั้งสองเหมือนกันเพียงพอ แสดงว่าได้เกิดการลู่เข้าก่อนวัยขึ้นแล้ว

สาเหตุของการเกิดการลู่เข้าก่อนวัยคือมีจำนวนประชากรน้อยเกินไปสำหรับปัญหานั้นๆ เทคนิคในการเพิ่มจำนวนประชากรสามารถแก้ปัญหาในส่วนนี้ได้ อีกสาเหตุหนึ่งของการเกิดการลู่เข้าก่อนวัยคือกลไกในการคัดเลือกสายพันธุ์ กล่าวคือถ้ามีการใช้กลยุทธ์หวักระติ ที่ซึ่งจะทำการคัดเลือกโครโมโซมที่มีค่าความเหมาะสมที่ดีที่สุด ส่งผลให้นำไปสู่ประชากรที่สืบสายพันธุ์เฉพาะจากประชากรกลุ่มเล็กๆ เท่านั้น วิธีแก้ปัญหานี้คือการเริ่มใหม่ (restart) กล่าวคือถ้าประชากรเกิดการลู่เข้าก่อนวัย โดยไม่มีสายพันธุ์ที่ดีขึ้น ให้ทำการเริ่มขั้นตอนทั้งหมดใหม่ การเริ่มใหม่นี้ทำให้มีโอกาสได้เริ่มต้นกับประชากรกลุ่มใหม่ๆ นั้นเอง

- **ความไม่อิสระของยีนในโครโมโซม** ในแง่มุมของระบบในโลกจริงแล้ว ตัวแปรหรือยีนในโครโมโซมหนึ่งๆ สามารถมีความสัมพันธ์ซึ่งกันและกันได้ ดังนั้นเราสามารถกล่าวได้ว่ายีนไม่มีความอิสระซึ่งกันและกัน ในกรณีนี้จะทำให้เกิดความยุ่งยากในขั้นตอนของการรวมตัวของยีน โดยปกติแล้วเราสามารถแก้ปัญหานี้ได้ด้วยการแยกยีนที่ไม่อิสระนี้ไว้เป็นกลุ่มๆ ในโครโมโซม กล่าวคือยีนกลุ่มนี้ไม่ควรได้รับการปรับเปลี่ยนในระหว่างปฏิบัติการทางสายพันธุ์ เช่นการทำการครอสโอเวอร์ เป็นต้น

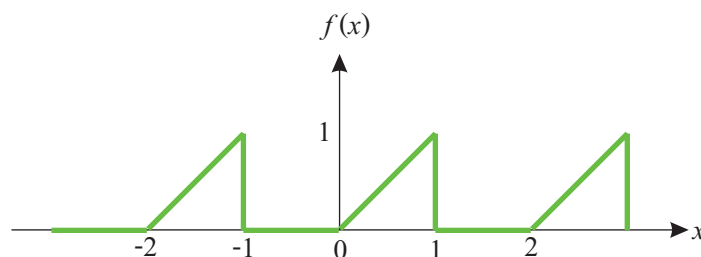
ถึงแม้ว่า GA จะเป็นวิธีที่มีประสิทธิภาพในการค้นหาคำตอบเพื่อให้ได้คำตอบที่ดีที่สุดในช่วงกว้าง ตัว GA เองก็ยังมีจุดอ่อนอยู่เช่นกัน โดยเฉพาะเมื่อนำ GA ไปใช้กับระบบที่มีการเชื่อมโยงกับสภาพแวดล้อมที่เป็นโลกจริง เนื่องจากคุณลักษณะของ GA ที่มีอยู่หลายขั้นตอนที่การคำนวณเป็นแบบสุ่ม ในบางครั้งจึงไม่ใช่เรื่องง่ายที่จะคาดหวังให้ผลลัพธ์จาก GA นั้นทำงานได้แล้วเสร็จโดยการกำหนดเวลาไว้อย่างแน่นอน ดังนั้น GA อาจจะไม่เหมาะกับระบบที่เป็นแบบเวลาจริง (real-time system) หรือระบบที่มีระยะเวลาในช่วงของการประมวลผลที่ค่อนข้างจำกัด อย่างไรก็ตาม GA ยังถือว่าเป็นเครื่องมือที่มีความชาญฉลาดในแง่ของปัญญาประดิษฐ์และเหมาะสมสำหรับ

ระบบที่มีข้อจำกัดแบบต่างๆ หรือระบบที่มีฟังก์ชันวัตถุประสงค์มากกว่าหนึ่ง รวมไปถึงคุณลักษณะสำคัญของ GA ที่สามารถเอาชนะปัญหาของการถูกล็อกโดยคำตอบวงแคบเฉพาะถิ่นที่เหมาะสมที่สุด ปัจจุบันจึงมีการนำเอา GA ไปประยุกต์ใช้ในงานต่างๆ มากมาย และยังนำไปใช้ร่วมกับวิธีการค้นหาคำตอบที่เป็นแบบปัญญาประดิษฐ์อื่นๆ เช่น ระบบเครือข่ายประสาทเทียม (neural network) เพื่อใช้ GA ในการค้นหาโครงสร้างของเครือข่ายที่เหมาะสมที่สุด หรือมีการนำ GA ไปใช้ในการปรับฟังก์ชันสมาชิกภาพของตัวแปรฟัซซี่ การนำ GA ไปประยุกต์ใช้จึงเป็นหัวข้อที่น่าสนใจเป็นอย่างยิ่ง



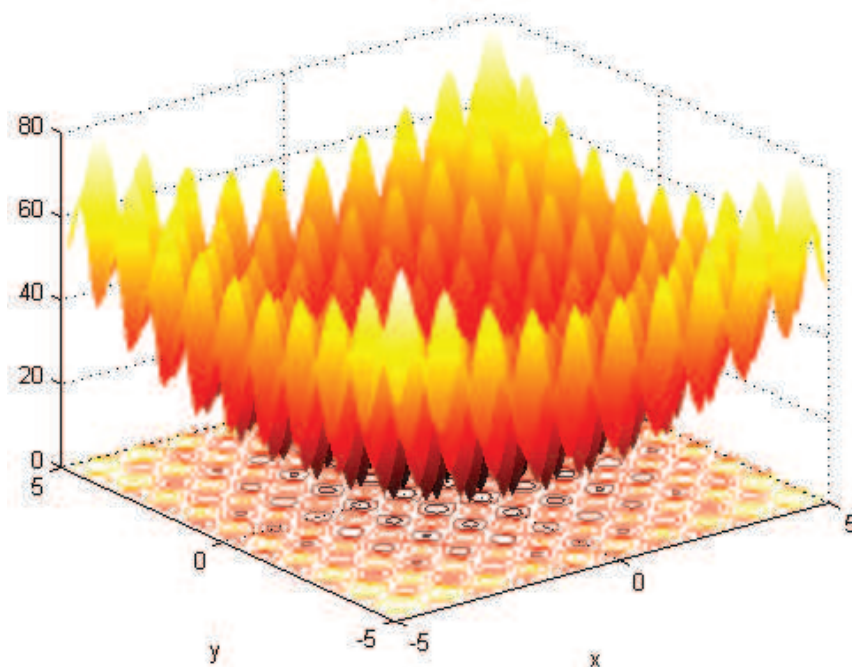
โจทย์คำถาม

- 3.1. จงออกแบบใช้ GA สำหรับค้นหาสัมประสิทธิ์อนุกรมฟูรีเยร์ของฟังก์ชัน $f(x)$ ในรูปที่ 3.33 อธิบายรายละเอียดการออกแบบ แสดงและวิเคราะห์ผลที่ได้



รูปที่ 3.33: ฟังก์ชันฟันเลื่อย

- 3.2. พิจารณาฟังก์ชันของ Rastrigin $f(x, y) = 20 + x^2 + y^2 - 10(\cos 2\pi x + \cos 2\pi y)$ ในรูปที่ 3.34 ที่ซึ่งประกอบไปด้วยจุดต่ำสุดทั้งแบบวงแคบเฉพาะถิ่น (local minima) และแบบวงกว้าง (global minima) จงออกแบบใช้ GA ในการหาค่าต่ำสุดแบบวงกว้างของฟังก์ชันดังกล่าว โดยทำการปรับพารามิเตอร์ต่างๆ ของ GA เพื่อเปรียบเทียบผลลัพธ์ที่ได้ ยกตัวอย่างเช่นการทำการครอสโอเวอร์ การทำมิวเทชัน การคัดเลือกสายพันธุ์ เป็นต้น



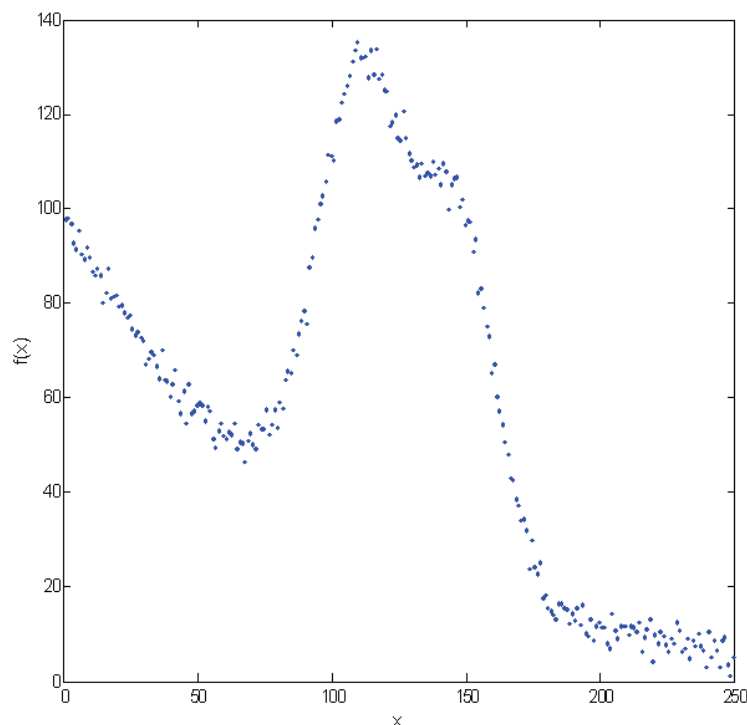
รูปที่ 3.34: ฟังก์ชัน Rastrigin

- 3.3. จงออกแบบ GA สำหรับทำการตัดแยกกลุ่มข้อมูล (clustering data) โดยทำการสุ่มค่าเวกเตอร์ (x_1, x_2) จำนวนทั้งหมด 100 เวกเตอร์ให้มีตำแหน่งอยู่กันเป็นกลุ่มๆ กำหนดค่าจำนวนกลุ่มสูงสุดที่จะทำการตัดแยกแล้วใช้ GA หาค่าตำแหน่งที่เป็นตัวแทนกลุ่มข้อมูล พร้อมทั้งระบุกลุ่มของเวกเตอร์ที่ทำการตัดแยกทั้งหมด

3.4. จงออกแบบใช้ GA ในการหาสัมประสิทธิ์การปรับเส้นโค้ง (curve fitting) ของข้อมูลแสดงในรูปที่ 3.35 โดยใช้ฟังก์ชันต่อไปนี้

$$f(x) = a_0 e^{-b_0 x} + a_1 e^{-\left(\frac{x-b_1}{c_1}\right)^2} + a_2 e^{-\left(\frac{x-b_2}{c_2}\right)^2}$$

- ใช้ฟังก์ชันที่เส้นโค้งข้างต้นในการสร้างข้อมูล ประกอบไปด้วยข้อมูลดั้งเดิมและชุดข้อมูลที่มีการเพิ่มสัญญาณรบกวน (ปรับเปลี่ยนขนาดของสัญญาณรบกวน)
- ใช้ GA ในการหาสัมประสิทธิ์ a_0, \dots, a_2 , b_0, \dots, b_2 และ c_1, \dots, c_2 ของชุดข้อมูลที่ได้สร้าง
- ปรับเปลี่ยนพารามิเตอร์ต่างๆ ของ GA เพื่อสังเกตผลการค้นหา
- วิเคราะห์และอภิปรายผลลัพธ์ที่ได้



รูปที่ 3.35: ข้อมูลสำหรับการปรับเส้นโค้ง

3.5. จงอธิบายการนำเอา GA ไปใช้ในการหาโครงสร้างที่เหมาะสมที่สุดของเครือข่ายประสาทเทียมแบบไปข้างหน้า (feed-forward network)

หมายเหตุ ดูรายละเอียดเพิ่มเติมในหัวข้อการเรียนรู้ด้วยเครือข่ายประสาทเทียม

- ทำการจำลองข้อมูลที่ต้องการให้เครือข่ายเรียนรู้
- ค้นหาค่าน้ำหนักประสาทและไบอัส
- ค้นหาโครงสร้างของเครือข่าย (ปกติพารามิเตอร์ของเครือข่ายประสาทเทียมจะยังไม่อยู่ในเงื่อนไขที่เหมาะสมที่สุด)

- J. Arabas and S. Kozdrowski. Applying an evolutionary algorithm to telecommunication network design. *IEEE Transactions on Evolutionary Computation*, 5(4):309--322, August 2001.
- A. Bajwa, T. Williams, and M.A. Stuchly. Design of broadband radar absorbers with genetic algorithm. In *IEEE International Symposium of Antennas and Propagation Society*, volume 4, pages 672--675, 2001.
- J.E. Baker. Adaptive selection methods for genetic algorithms. In *Proceeding of an International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates (Hillsdale), 1985.
- W.A. Bedwani and O.M. Ismail. Genetic optimization of variable structure pid control systems. In *ACS/IEEE International Conference on Computer Systems and Applications 2001*, pages 27--30, 2001.
- G. Lo Bosco. A genetic algorithm for image segmentation. In *Proceedings of 11th International Conference on Image Analysis and Processing 2001*, pages 262--266, 2001.
- O. Castillo, O. Montiel, R. Sepulveda, and P. Melin. Application of a breeder genetic algorithm for system identification in an adaptive finite impulse response filter. In *Proceedings of The 3rd NASA/DoD Workshop on Evolvable Hardware 2001*, pages 146--153, 2001.
- A.J. Chipperfield, P.J. Fleming, and H. Pohlheim. A genetic algorithm toolbox for matlab. In *Proceedings of International Conference on System Engineering*, Coventry, UK, 1994.
- O. Cordon, F. Herrera, and P. Villar. Generating the knowledge base of a fuzzy rule-based system by the genetic learning of the database. *IEEE Transactions on Fuzzy Systems*, 9(4):667--674, August 2001.
- C. Goh and Y. Li. Ga automated design and synthesis of analog circuits with practical constraints. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 1, pages 170 --177, 2001.
- D.E. Goldberg. *Genetic Algorithm in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing, 1989.
- D.E. Goldberg. Genetic and evolutionary algorithms come of age. *Communication of the ACM*, 37(3): 113--119, March 1994.

- J.J. Grefenstette. *A User's Guide to GENESIS v5.0*. Naval Research Laboratory, Washington D.C., 1990.
- J.B. Grimbleby. Automatic analogue circuit synthesis using genetic algorithms. In *IEE Proceedings on Circuits, Devices and Systems*, volume 147, pages 319--323, December 2000.
- H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor: the University of Michigan Press, Michigan, 1975.
- F. Hussein, N. Kharma, and R. Ward. Genetic algorithms for feature selection and weighting: A review and study. In *Proceedings of 6th International Conference on Document Analysis and Recognition 2001*, pages 1240--1244, 2001.
- C.Z. Janikow and Z. Michalewicz. An experimental comparison of binary and floating point representations in genetic algorithms. In *Proceedings of 4th International Conference on Genetic Algorithms*, pages 31--36, 1991.
- E. Jones, P. Runkle, N. Dasgupta, L. Couchman, and L. Carin. Genetic algorithm wavelet design for signal classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):890--895, August 2001.
- M. Kezunovic and Y. Liao. Fault location estimation based on matching the simulated and recorded waveforms using genetic algorithms. In *IEE International Conference on Developments in Power System Protection*, pages 399--402, 2001.
- K. Sun Kim and Z. Byoung-Tak. Evolutionary learning of web-document structure for information retrieval. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 2, pages 1253--1260, 2001.
- C. Chi Kin, T. Hung Tat, L. Tong, and L. Tze Kin. Medical image registration and model construction using genetic algorithms. In *Proceedings of International Workshop on Medical Imaging and Augmented Reality*, pages 174--179, 2001.
- E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley and Sons Inc., 1999.
- P. Kumsawat, K. Attakitmongcol, A. Srikaew, and S. Sujitjorn. Wavelet-based image watermarking using the genetic algorithm. In *Lecture Notes in Computer Science*, volume 3215, pages 643--649. Springer-Verlag, October 2004a.
- P. Kumsawat, K. Attakitmongcol, A. Srikaew, and S. Sujitjorn. Wavelet-based image watermarking using genetic algorithm. In *The 8th Knowledge-based Intelligent Information and Engineering Systems International Conference*, Wellington, New Zealand, September 2004b.
- P. Kumsawat, K. Attakitmongcol, and A. Srikaew. A new approach for optimization in image watermarking by using genetic algorithm. In *IEEE Transaction on Signal Processing*, volume 53, pages 4707--4719, December 2005a.
- P. Kumsawat, K. Attakitmongcol, and A. Srikaew. A new approach for optimization in wavelet-based image watermarking by using genetic algorithm. In *Proceeding of the 23rd IASTED International Multi-Conference Artificial Intelligence and Applications*, volume 1, pages 328--332, Innsbruck, Austria, February 2005b.

- S.S. Lam. A genetic fuzzy expert system for stock market timing. In *Proceedings of the 2001 Congress on Evolutionary Computation*, volume 1, pages 410--417, 2001.
- K.F. Man, K.S. Tang, S. Kwong, and W.A. Halang. *Genetic Algorithms for Control and Signal Processing*. Springer-Verlag, London, 1997.
- G. Manganaro. Genetic algorithms for vlsi design, layout, and test automation: Reviews. *IEEE Circuits and Devices Magazine*, 16(2):34--34, March 2000.
- MathWorks. *MATLAB Help: Genetic Algorithm and Direct Search Toolbox*. Mathworks Inc, 2004.
- Mathworks. *MATLAB User's Guide*. Mathworks Inc, 1991.
- P. Meesad and G.G. Yen. A hybrid intelligent system for medical diagnosis. In *Proceedings of International Joint Conference on Neural Networks*, volume 4, pages 2558 --2563, 2001.
- P. Melin and O. Castillo. Intelligent control of nonlinear dynamical systems with a neuro-fuzzy-genetic approach. In *Proceedings of International Joint Conference on Neural Networks*, volume 1, pages 515--520, 2001.
- H. Min-Huang, C. Ming-Chun, C. Yue-Shan, and Y. Shyan-Ming. A ga-based dynamic personalized filtering for internet search service on multi-search engine. In *Canadian Conference on Electrical and Computer Engineering*, volume 1, pages 271--276, 2001.
- M. Minami, H. Suzuki, J. Agbanhan, and T. Asakura. Visual servoing to fish and catching using global/local ga search. In *Proceedings of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, volume 1, pages 183--188, 2001.
- Y. Mitsukura, N. Fukumi, and N. Akamatsu. A detection method of face regions in color images by using evolutionary computation. In *Proceedings of IJCNN International Joint Conference on Neural Networks*, volume 3, pages 2253--2257, 2001.
- R. Moller and R. Zeipelt. Automatic segmentation of 3d-mri data using a genetic algorithm. In *Proceedings of International Workshop on Medical Imaging and Augmented Reality*, pages 278--281, 2001.
- Z.Z. Nick and P. Themis. Web search using a genetic algorithm. *IEEE Internet Computing*, 5(2):18--26, March-April 2001.
- E. Poirier, M. Ghribi, and A. Kaddouri. Loss minimization control of induction motor drives based on genetic algorithms. In *IEEE International Electric Machines and Drives Conference*, pages 475--478, 2001.
- A. Prügel-Bennett. The mixing rate of different crossover operators. In W.N.Martin and W. Spears, editors, *Foundations of Genetic Algorithms 6*. Morgan Kaufmann, 2001.
- A. Prügel-Bennett. Finite population effects for ranking and tournament selection. *Complex Systems*, 12(2):183--205, 2000.
- A. Prügel-Bennett and J.L. Shapiro. The dynamics of a genetic algorithm for simple random ising systems. *Physica D*, 104:75--114, 1997.
- A. Prügel-Bennett and J.L. Shapiro. An analysis of genetic algorithms using statistical mechanics. *Physical Review Letters*, 72(9):1305--1309, 1994.

- A. Rogers and A. Prügel-Bennett. Evolving populations with overlapping generations. *Theoretical Population Biology*, 57(2):121--129, 2000.
- A. Rogers and A. Prügel-Bennett. Modelling the dynamics of steady-state genetic algorithms. In W. Banzhaf and C. Reeves, editors, *Foundations of Genetic Algorithms 5*, pages 57--68. Morgan Kaufmann, 1999.
- J.A. Smith. Designing biomorphs with and interactive genetic algorithm. In *Proceedings of 4th International Conference on Genetic Algorithm*, 1991.
- Arthit Srikaew. Genetic algorithm - part ii case study: How to search for fourier coefficients (in thai). *Suranaree Journal of Science and Technology*, 9(2):139--146, April-June 2002.
- D. Srinivasan, C. Rucy Long, and P. Young Peng. Hybrid fuzzy logic-genetic algorithm technique for automated detection of traffic incidents on freeways. In *Proceedings of 2001 IEEE Intelligent Transportation Systems*, pages 352 --357, 2001.
- Y.C. Tang. *TOLKIEN Reference Manual*. Department of Computer Science, Chinese University of Hong Kong, 1994.
- B. Thomas. *User's Guide for GENESys*. Department of Computer Science, University of Dortmund, 1992.
- A. Visioli. Optimal tuning of pid controllers for integral and unstable processes. In *IEE Proceedings of Control Theory and Applications*, volume 148, pages 180--184, March 2001.
- D.S. Weile and E. Michielssen. The control of adaptive antenna arrays with genetic algorithms using dominance and diploidy. *IEEE Transactions on Antennas and Propagation*, 49(10):1424--1433, October 2001.
- Y.K. Wong, T.S. Chung, and W.M. Lai. Application of genetic algorithm in reactive power/voltage control-problem. In *International Conference on Advances in Power System Control, Operation and Management*, volume 2, pages 486--490, 2000.
- A. H. Wright. *Foundations of Genetic Algorithms*, chapter Genetic Algorithms for Real Parameter Optimization, pages 205--218. Morgan Kaufmann, 1991.
- Y. Yamazaki, V. Aiyoshin, J. Krasilnikova, and I. Krasilnikov. Adaptive-intelligent control by neural-network and genetic-algorithm systems and its application. In *Proceedings of Second International Conference on Knowledge-Based Intelligent Electronic Systems*, volume 3, pages 230--239, 1998.
- Q. Yang, F. Lin, G. Zhang, and J. Jiang. The application of computational intelligence to fourier transformation. In *The 7th IEEE International Conference on Electronics, Circuits and Systems*, volume 1, pages 178--181, 2000.
- S. Yong-Hua and M.R. Irving. Optimisation techniques for electrical power systems. ii. heuristic optimisation methods. *Power Engineering Journal*, 15(3):151--160, June 2001.